



## ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO TÉCNICO DE TELECOMUNICACIÓN,  
ESPECIALIDAD EN SONIDO E IMAGEN

Título del proyecto:

“OPTIMIZACIÓN DE LA ECUACIÓN DE  
CALIBRACIÓN DE UN SISTEMA DE EYE-  
TRACKING VÍDEO OCULOGRÁFICO”

Maria Villanueva Almirantearena

Tutores: Arantxa Villanueva Larre

Juan José Cerrolaza Martínez

Pamplona, 29/07/2010

## ÍNDICE

<b>INTRODUCCIÓN</b>	<b>4</b>
<b>1 Visión general de los sistemas de eye-tracking</b>	<b>5</b>
<b>1.1 Descripción e historia de los sistemas de eye-tracking</b>	<b>5</b>
1.1.1 Historia de los sistemas más antiguos	5
1.1.2 Clasificación de las técnicas de eye-tracking	7
1.1.3 Fisiología del ojo humano	9
<b>1.2 Sistemas video oculográficos (VOG)</b>	<b>10</b>
1.2.1 Configuraciones posibles	12
1.2.2 Sistemas basados en modelos matemáticos y sistemas basados en ecuaciones polinómicas.	14
<b>2 Descripción del sistema de eye-tracking utilizado</b>	<b>17</b>
<b>2.1 Calibración</b>	<b>18</b>
2.1.1 Análisis de regresión lineal	19
<b>2.2 Inconvenientes que presenta el sistema</b>	<b>20</b>
<b>3. Optimización del proceso de calibración de los sistemas video-oculográficos</b>	<b>22</b>
<b>3.1 Ecuación de calibración de partida</b>	<b>22</b>
<b>3.2 Error que presenta el sistema con el movimiento de la cabeza</b>	<b>23</b>
<b>4 SHAKTI</b>	<b>24</b>
<b>4.1 Tratamiento de los datos</b>	<b>24</b>
4.1.1 Filtrado de los datos	24
4.1.2 Cálculo de los coeficientes de la ecuación de calibración	25
<b>5 Descripción y caracterización de FaceAPI</b>	<b>27</b>
<b>5.1 Funcionamiento de FaceAPI</b>	<b>27</b>
<b>5.2 Caracterización de FaceAPI</b>	<b>28</b>
<b>5.2.1 Análisis de los datos</b>	<b>28</b>
5.2.1.1 Ejes	29
5.2.1.2 Magnitud y unidades de medida de la posición y rotación de la cabeza	30
5.2.1.2.1 Desplazamientos máximos	31
5.2.1.3 Variación de los datos con la cabeza quieta (Ruido de FaceAPI)	31
5.2.1.3.1 Con fotos de caras	31
5.2.1.3.2 Con usuarios reales	32
<b>5.3 Tiempos de captura</b>	<b>33</b>
5.3.1 Estudio de los tiempos de adquisición de FaceAPI	33
<b>5.4 Problemas en la adquisición de datos</b>	<b>34</b>
5.4.1 Funcionamiento a tiempo real	34
5.4.2 Desplazamiento en el eje Z.	40
<b>5.5 Conclusiones</b>	<b>45</b>
<b>6 Estudio experimental</b>	<b>46</b>
<b>6.1 Posibles movimientos a estudiar.</b>	<b>46</b>
6.1.1 Filtrado de movimientos posibles a estudiar	47
6.1.2 Alcance del estudio	47
<b>6.2 Sistema utilizado para las pruebas</b>	<b>48</b>
6.2.1 Alcance del recorrido por el eje Z	48
6.2.1.1 Posición de los iluminadores	49

6.3	Colocación del resto de elementos	50
6.4	Descripción de las pruebas	50
6.4.1	Protocolo experimental	51
6.5	Sincronización entre Shakti y FaceAPI	52
6.6	Conclusiones	53
7	Análisis de los resultados	54
7.1	Presentación de los resultados	54
7.1.1	Cálculo del $PoR$ para cada posición	54
7.1.2	Trayectoria del $PoR$ estimado	57
7.1.3	Error entre el $PoR$ real y el estimado	59
7.1.4	Dirección del error	64
7.1.5	Comportamiento medio en cada posición.	65
7.1.6	Correlación entre FaceAPI y el error en $Y$ .	67
7.1.7	Conclusiones	69
7.2	Propuesta de ecuaciones de mejora	69
7.2.1	Normalización de $V_y$	70
7.2.2	Añadir un término de corrección a la ecuación $PoRY$ .	74
7.2.2.1	Corrección fija de la ecuación de calibración	76
7.2.2.2	Corrección de la ecuación de calibración individual	78
7.2.3	Conclusiones	78
7.3	Aplicación de las nuevas ecuaciones	78
7.3.1	Error medio de cada posición	79
7.3.2	Estudio estadístico de las ecuaciones corregidas	82
7.3.3	Conclusiones	83
7.4	Estudio de las nuevas ecuaciones ante desplazamientos en el eje $X$ e $Y$	84
7.4.1	Datos de usuarios reales	84
7.4.2	Datos provenientes de un modelo matemático	93
7.4.3	Conclusiones	97
8	Conclusiones	99
9	Líneas futuras	101
	Bibliografía	102

## **INTRODUCCIÓN**

El presente proyecto tiene como objetivo optimizar la ecuación de calibración de un sistema de eye-tracking video oculográfico basado en ecuaciones polinomiales para dotar al sistema de mayor robustez ante desplazamientos del usuario.

Una de las limitaciones de los sistemas video-oculográficos que utilizan una sola cámara es la imposibilidad de determinar la localización en el espacio 3D de los objetos; imposible sin un sistema calibrado y el sistema de partida que utilizaremos no podrá medir la distancia del usuario a la cámara. Por ello recurriremos a otra herramienta (FaceAPI) como posible solución. FaceAPI es un software de seguimiento de caras que permite calcular la posición y orientación de la cabeza en 3D a través de una webcam.

Primero se va intentar hallar la posición de la cabeza del usuario a través del uso de un software de seguimiento de caras: FaceAPI. Esta información será incorporada a la ecuación de calibración para modificarla ante desplazamientos del usuario. Es decir, se intentará añadir un nuevo término a la ecuación de calibración que dependa directamente de la posición y orientación espacial de la cabeza calculada por este software. Dado que FaceAPI ha sido recientemente incorporado a la Universidad y se desconoce su funcionamiento será necesaria su previa caracterización. Una vez conocido el funcionamiento del software se intentará extraer la información ofrecida por el mismo.

Después de estudiar el comportamiento de FaceAPI se realizará un estudio de la cuantía del error presentado por el sistema de eye-tracking ante desplazamientos de los usuarios controlados y al mismo tiempo cuantificados por FaceAPI. Es decir, se va a realizar un estudio experimental con diferentes usuarios en el que observemos la tendencia del error de la estimación del *PoR* ante desplazamientos controlados. Así se intentará compensar este error con la información de FaceAPI.

La inclusión de la información de FaceAPI en el sistema de eye-tracking se realizará sincronizando ambos sistemas. Para ello en cada muestra de información de ambos programas se incluirá información del tiempo de captura.

## **1 Visión general de los sistemas de eye-tracking**

### **1.1 Descripción e historia de los sistemas de eye-tracking**

#### **1.1.1 Historia de los sistemas más antiguos**

Los seres humanos tenemos una gran cantidad de comportamientos asociados a los movimientos del ojo y a la dirección de la mirada: como por ejemplo la lectura, la orientación, el entretenimiento viendo la televisión, etc. El estudio del comportamiento visual ha contribuido a la comprensión del sistema visual y los mecanismos relacionados con la percepción y funcionamiento cerebral para el desarrollo de sistemas eye-tracking.

El concepto de eye-tracking hace referencia al conjunto de tecnologías que permiten monitorizar o evaluar el punto donde se fija la mirada, es decir, el movimiento del ojo en relación con la cabeza. Estos sistemas tienen un gran potencial de aplicación en varias disciplinas y áreas de estudio, desde el marketing y publicidad hasta la investigación médica o la psicolingüística.

Esta tecnología puede parecer reciente, pero la labor de estudiar el movimiento ocular comenzó un siglo atrás como advierten Jacob y Karn en su estudio [26]. Últimamente es notorio el creciente interés por la investigación y desarrollo de este tipo de sistemas, probablemente sea debido a la proliferación de soluciones comerciales a precios relativamente asequibles. Hoy en día existen diferentes sistemas en el mercado (Iriscom, Tobii, etc.). Los sistemas de Iriscom, por ejemplo, incluyen el hardware necesario (cámara e iluminadores) y un software específico a precios cercanos a los 8000-11000€. Sin embargo, los avances tecnológicos han hecho posible el desarrollo de sistemas más baratos. Actualmente se está trabajando en sistemas que permiten el acceso al público mayoritario [4], [21].

Los primeros estudios sobre el movimiento de los ojos se realizaron mediante la observación visual directa (o con ayuda de telescopios o espejos) o introspección. En 1879, Javal [27] observó los movimientos del ojo al leer, manteniendo un ojo cerrado y junto a este un micrófono que detectaba sus movimientos. Delabarre [28], también estudió estos movimientos sellándolos con un quimógrafo. Más adelante, en 1900, Huey [29] construyó un sistema de eye-tracking utilizando una especie de lente de contacto con un agujero para el usuario. La lente estaba conectada a un puntero de aluminio que se movía en respuesta al movimiento del ojo.

Delabarre y Huey fueron capaces de obtener una primera visión muy valiosa de la función y características de los movimientos oculares, pero sus métodos fueron criticados por resultar demasiado invasivos. Para superar estos inconvenientes Dodge y Cline [30] en 1901 registraron el movimiento de los ojos de manera más precisa y no invasiva. Dodge [31] emitía líneas de luz a los ojos y grababa su reflejo mediante la fotografía (gran aportación hasta los siguientes avances en los 70 con la era digital). Esta fue la primera investigación del movimiento ocular utilizando la reflexión de la córnea, método que hoy en día se sigue utilizando.

Gracias a la evolución de la década de 20 en laboratorios Chicago y Standfor se iniciaron los primeros registros de movimientos oculares en tres dimensiones capturando dos imágenes del ojo simultáneamente. Más adelante, se consiguió descomponer la reflexión de los haces de luz de un ojo en sus componentes horizontal y vertical. Buswell [32] en 1935 utilizó esta descomposición de la reflexión del haz para medir las rutas de análisis de un sujeto al observar una imagen.

En la década de los 60 se realizan investigaciones utilizando otra vez técnicas invasivas para mejorar la exactitud de las técnicas anteriores. En este caso, lentes de contacto junto con espejos o incluso rollos de alambre, que utilizaron investigadores como Yarbus [34] o Fender [33].

En la década de 1970, la investigación de seguimiento del ojo se expandió rápidamente, en particular la investigación del movimiento del ojo en la lectura (Rayner [35]). Esta expansión se produjo también por las nuevas técnicas que empezaron a utilizarse, como el escaneo del ojo mediante el uso de una cámara. Con esta técnica se conseguía alto contraste entre el iris y la esclerótica facilitando su separación.

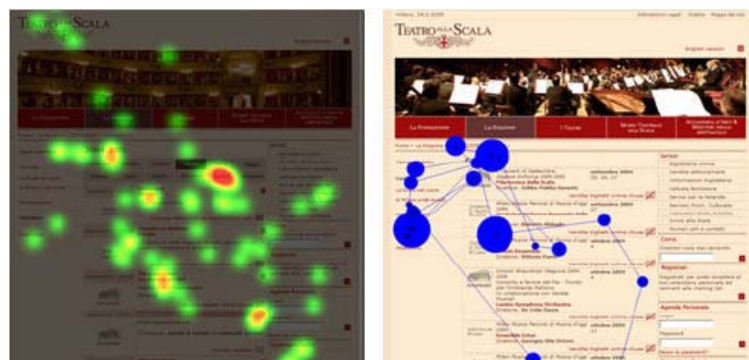
De esta década destacan investigadores como Taylor S. E. [36] quien inventó un dispositivo mecánico para traducir los movimientos del ojo en permanentes o Merchant y Morrisette [24] quienes a través de su oculómetro calculaban la posición del centro de la pupila y los reflejos en la imagen.

Todos los métodos anteriores miden el movimiento de los ojos respecto a la cabeza, por lo que los investigadores debían de utilizar sistemas para permanecer con la cabeza inmóvil. Esto fue obviado en esta década al utilizar dos características ópticas del ojo en movimiento, ya que los reflejos de los haces de luz permanecen invariantes ante traslaciones de la cabeza pero varían con rotaciones del ojo (según el estudio de Merchant).

Actualmente la tecnología de eye-tracking principalmente sigue siendo reservada a tareas de investigación y de desarrollo de productos, aunque se espera que su evolución sea muy rápida en los próximos años, hasta el punto de poder llegar a convertirse en interfaz de uso cotidiano en dispositivos de consumo. Según los expertos, hoy en día nos encontramos en la cuarta era de los sistemas de eye-tracking caracterizada por el gran número de nuevas aplicaciones de las cuales las más representativas serán expuestas a continuación.

Atendiendo al criterio de Andrew T. Duchowsky [5] se pueden distinguir dos vertientes de aplicación de sistemas de eye-tracking: diagnósticas (pasivas, de análisis y diagnóstico) e interactivas (como medio de interacción con el computador).

En el primero de los casos, el sistema de eye-tracking proporciona la evidencia objetiva y cuantitativa de los procesos relacionados con la visión y la percepción, midiendo el movimiento de los ojos y la atención al realizar otra tarea independiente. Es decir, funciona como herramienta de evaluación: la información extraída ya no requiere de un análisis y respuesta en tiempo real por el sistema sino que es registrada para su posterior análisis e interpretación. Esta técnica es aplicada por ejemplo en campos de marketing o para diseños de páginas web, mediante mapas de calor que miden las zonas que llaman más la atención al observar una escena determinada (ver fig. 1.1).



**Fig. 1.1** Densidad de fijaciones en una página web

Cuando se utiliza como medio interactivo sirve como un dispositivo de entrada sustituyendo al ratón. Aunque la precisión del sistema de eye-tracking como dispositivo de entrada dista de la de otros, como el ratón, puede tener numerosas aplicaciones prácticas, tales como su uso en entornos de realidad virtual o por usuarios con discapacidad motriz (como la esclerosis múltiple). Pero además, incluso en determinadas operaciones como la selección de objetos de la interfaz, la mirada puede resultar más rápida que la acción mediante el ratón.

### 1.1.2 Clasificación de las técnicas de eye-tracking

Los sistemas de eye-tracking se pueden dividir en tres categorías por las metodologías de medición de movimientos oculares que impliquen: electro-oculografía (EOG), lentes de contacto/ bobinas de búsqueda y foto-oculografía (FOG) o video-oculografía (VOG).

Una de las características de los sistemas eye-tracking requerida para la comodidad del usuario, como hemos visto en las diferentes técnicas utilizadas durante la historia, es el carácter no invasivo de los sistemas. De esta forma podríamos agrupar las tres categorías de las metodologías antes mencionadas en dos: sistemas invasivos y no invasivos.

#### a) Sistemas invasivos:

Entre los sistemas invasivos que requieren el contacto físico con el sujeto encontramos los sistemas basados en lentes de contacto/bobinas de búsqueda y sistemas basados en electro-oculografía (EOG). Generalmente estos sistemas realizan mediciones de mayor precisión.

##### a.1) Sistemas basados en lentes de contacto:

Esta es una de las técnicas más precisas que implica el contacto físico con el ojo a través de un mecanismo basado en lentes de contacto, pero inevitablemente estos sistemas resultan muy incómodos para los usuarios.

Esta técnica implica asignar un objeto de referencia mecánico u óptico para ser montado en la lente de contacto que luego se coloca directamente en el ojo, como por ejemplo fósforos que reflejan, diagramas de línea o bobinas inductoras. El método principal emplea bobinas de búsqueda y el movimiento se mide a través de un campo electromagnético. A pesar de que es el método más preciso (con una precisión de 5-10 arco segundos en un rango limitado de 5 °) es el método más invasivo.

Estas técnicas no parecen ofrecer una buena solución a la integración a la tecnología de eye-tracking. A su vez, los problemas de salud derivados de los campos magnéticos de alta frecuencia no han sido resueltos todavía existiendo ciertas dudas sobre su efecto en el organismo humano. Sin embargo, muchos expertos expresan sus reservas sobre esta tecnología aún y cuando se lograsen solucionar estos problemas por tratarse de una técnica excepcionalmente invasiva.

##### a.2) Electro-oculografía (EOG)

Esta técnica fue una de las más utilizadas hace unos cuarenta años, consiste en la medida de diferencias de potencial eléctricas en la piel detectadas por electrodos colocados alrededor del ojo. Esta técnica en principio ofrecería gran libertad de movimientos, pero al necesitar detectar la posición de la cabeza también se pierde esta propiedad. Estas diferencias de potencial se producen con movimientos de los ojos pero no son constantes y sus variaciones hacen difícil de usar EOG para medir los movimientos oculares lentos y la detección de

dirección de la mirada. Sin embargo, es una técnica muy sólida para medir los movimientos oculares sacádicos asociados con cambios de la mirada y la detección de parpadeos. Además, esta técnica requiere una potencia de cálculo muy baja y se puede utilizar en ausencia de iluminación.

### b) Sistemas no-invasivos:

Entre los sistemas no invasivos encontramos aquellos basados en técnicas de video o foto oculografía. El carácter no invasivo de estos sistemas hace que hoy en día sean los más utilizados. En este apartado serán introducidos los sistemas de video-oculografía, no obstante al tratarse de la tecnología que utilizaremos en este proyecto, en el apartado siguiente serán desarrollados en detalle.

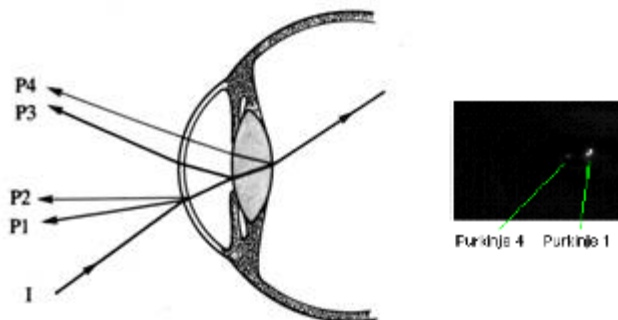
#### b.1) Foto oculografía (POG) y video-oculografía (VOG)

Esta técnica incorpora una cámara o dispositivo de adquisición de imágenes para tratar de determinar el movimiento de los ojos utilizando las imágenes obtenidas por dicho dispositivo. Para ello, se analiza el comportamiento de características del ojo humano como el centro de la pupila, el contorno del iris o las reflexiones de la córnea e intentando establecer una relación entre estas y el punto donde se fija la mirada (*PoR*).

Estos sistemas están basados en la captura de imágenes por lo que requieren una iluminación del usuario adecuada. Algunos de estos sistemas cuentan con una o varias fuentes de iluminación aunque también existen sistemas que funcionan solamente con la luz visible.

En el caso de utilizar iluminadores, generalmente la luz emitida por estos es infrarroja. Esta se refleja en los ojos y es capturada por la cámara de vídeo. La información capturada se analiza para extraer la rotación de los ojos en función de los cambios de posición de las reflexiones.

Distinguimos diferentes eye-trackers por la característica del ojo que analizan. Aprovechando la reflexión del ojo, hay eye-trackers que emplean el primer reflejo de la córnea (primera imagen de Purkinje). Esta es la reflexión más fácil de detectar pero, también se producirán más reflejos de los cuales el primero y cuarto serán los más relevantes.



**Fig.1.2** Reflejos de la córnea.

No obstante en este proyecto utilizaremos uno de los métodos oculográficos que mejor funcionan el cual combina este primer reflejo de la córnea (1ª imagen de Purkinje) y la información del centro de la pupila.

Así mismo, las técnicas de eye-tracking mediante video-oculografía podrían dividirse en función de su objetivo:

- Técnicas que miden los movimientos del ojo
- Técnicas que determinan el punto exacto donde se fija la mirada, o cálculo del *PoR*.



La primera técnica determina la rotación del ojo, esto es, la orientación del ojo respecto a la cabeza. En la segunda, se calcula el punto a donde el sujeto mira en el espacio; para tal fin será necesario el seguimiento independiente de la posición de la cabeza (en 3D) además de conocer la rotación del ojo.

Entre los sistemas basados en video-oculografía distinguimos los basados en modelos matemáticos y los basados en modelos polinómicos. (Ver apartado 2.2). Cualquiera de estos sistemas podemos diseñarlo para ir montados en la cabeza, como un casco, esta configuración consta de una cámara para el seguimiento del movimiento del ojo y otra cámara capturando la escena a la que el usuario mira, o como sistema remoto aunque en muchos casos habría que mantener la cabeza quieta utilizando un reposa cabezas debido a la falta de robustez de estos sistemas ante desplazamientos de la cabeza). (Ver apartado 1.2).

Ambas soluciones conllevan a que el sistema pierda el carácter no invasivo, es decir, perdería la cualidad que hace que estos sistemas sean los más utilizados.

### 1.1.3 Fisiología del ojo humano

Sin lugar a dudas, el principal elemento de todo sistema de eye-tracking es el ojo humano, uno de los órganos más complejos y fascinantes que existen. En este apartado dedicaremos unas líneas a describir de forma general este preciso ‘instrumento óptico’.

El ojo es un órgano que detecta la luz y es la base del sentido de la vista. Se compone de un sistema sensible a los cambios de luz, capaz de transformar éstos en impulsos eléctricos. El ojo en su conjunto es una estructura esférica de aproximadamente 2,5cm de diámetro con un marcado abombamiento sobre su superficie delantera.

En la porción anterior del ojo se encuentran dos pequeños espacios, la cámara anterior que está situada entre el iris y la córnea y la cámara posterior que se ubica entre la córnea y el cristalino. Estas cámaras están llenas de un líquido que se llama humor acuoso, cuyo nivel de presión llamado presión intraocular es relevante para el correcto funcionamiento del ojo.

La capa externa del ojo o esclerótica tiene una función protectora, cubre unos cinco sextos de la superficie ocular. Esta se prolonga en la parte anterior con la córnea, superficie que posibilita el paso de los rayos de luz. La capa media o úvea tiene tres partes diferenciadas: la coroides la cual reviste las tres quintas partes posteriores del globo ocular; el cuerpo ciliar, formado por los procesos ciliares, y el iris, en la parte frontal del ojo el cual regula el diámetro de un diafragma denominado pupila, controlando así la cantidad de luz que entra al ojo permitiendo diferentes condiciones de iluminación. Y por último un tejido interno, la retina, sensible a la luz.

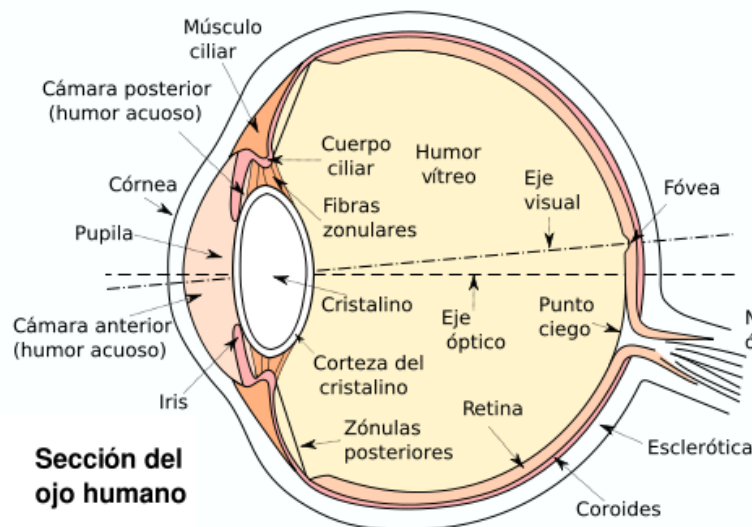
La luz penetra a través de la pupila, atraviesa el cristalino y se proyecta sobre la retina formando una imagen invertida, donde se transforma gracias a unas células fotorreceptoras en impulsos nerviosos que son trasladados a través del nervio óptico al cerebro. Estas células se clasifican funcionalmente en conos y bastones. Los primeros son sensibles a luz cromática y el brillo (la luz del día) y los bastones responden a la luz de la noche (visión purkinje). La retina tiene una pequeña mancha de color amarillo, llamada mácula lútea y en su centro se encuentra la fovea central, la zona del ojo con mayor agudeza visual. La capa sensorial de la fovea se compone sólo de células con forma de conos, mientras que en torno a ella también se encuentran células con forma de bastones. Según nos alejamos del área sensible, las células con forma de cono se vuelven más escasas y en los bordes exteriores de la retina sólo existen las células con forma de bastones.

Para que los rayos de luz que penetran en el ojo se puedan enfocar en la retina, se deben refractar. La cantidad de refracción requerida depende de la distancia del objeto al observador (un objeto distante requerirá menos refracción que uno más cercano). La mayor parte de la refracción ocurre en la córnea, que tiene una curvatura fija. Otra parte de la refracción requerida se da en la lente del cristalino. El cristalino está constituido por fibras transparentes dispuestas en capas conectadas con el músculo ciliar. El cuerpo ciliar que rodea al cristalino se encarga de cambiar de la longitud focal de la lente, aumentando o disminuyendo de esta forma su capacidad de refracción.

Por detrás de la lente, el cuerpo principal del ojo está lleno de una sustancia transparente y gelatinosa (el humor vítreo) encerrado en un saco delgado que recibe el nombre de membrana hialoidea. La presión del humor vítreo mantiene distendido el globo ocular.

Al intervalo de valores de longitudes de onda desde los 350nm a los 780nm corresponden con la luz visible. La distribución de la sensibilidad del ojo a las distintas longitudes de onda tiene forma de campana con un valor máximo para los conos en torno a los 600 nm y un máximo para los bastones en torno a los 500 nm.

Para aquellos sistemas de eye-tracking cuyo objetivo es determinar la línea o dirección de visión de un sujeto, es importante diferenciar entre eje óptico y eje visual. El eje óptico es una línea que pasa por el centro del sistema óptico del ojo. El eje visual es una línea que une el punto de fijación en el centro del objeto observado, el punto nodal del sistema óptico, ubicado cerca de la superficie posterior del cristalino, y la fovea central. El eje óptico no coincide exactamente con el eje visual porque el cristalino se encuentra ligeramente descentrado hacia abajo, lo que no altera la función visual. El desfase entre ambos ejes es del orden de  $5 \pm 2^\circ$ .



**Fig.1.3** Sección del ojo humano

## 1.2 Sistemas video oculográficos (VOG)

Los elementos básicos de un sistema de eye-tracking basados en video-oculografía son como ya hemos comentado anteriormente las fuentes de iluminación y el sistema de adquisición de imágenes empleado. En los últimos años han sido desarrollados diversos algoritmos utilizando sistemas video oculográficos. Aunque todos ellos tengan la misma base

admiten múltiples variaciones como el tipo de iluminación, el número de iluminadores o cámaras utilizadas, la localización de cada uno de los elementos, etc.

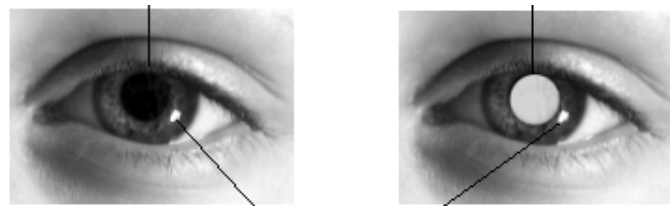
Hay dos tipos de iluminación de uso común utilizados para el seguimiento de la mirada: la luz visible y la infrarroja. La utilización de imágenes del espectro visible es una actitud pasiva la cual captura la luz ambiental reflejada en el ojo. La mejor característica para realizar un seguimiento con este tipo de iluminación es el contorno entre el iris y la esclerótica conocido como el limbo (fig. 1.5 dcha). Las tres características más relevantes del ojo son la pupila, el iris, y la esclerótica. La utilización de este tipo de iluminación se complica por el hecho de que la luz de ambiente puede contener varios componentes de luz especulares. En cambio, al utilizar iluminación infrarroja se elimina la reflexión especular no controlada, y permite que la iluminación de los ojos sea uniforme, no siendo perceptible para el usuario. Un beneficio adicional de la proyección infrarroja es que la característica más fuerte, en cuanto a los contornos de la imagen, es la pupila y no el limbo (ver fig. 1.5 izda); la esclerótica y el iris reflejan la luz infrarroja, en cambio, la luz visible solamente la esclerótica. El seguimiento del contorno de la pupila es preferible dado que el contorno de la pupila es más pequeño y más definido que el del limbo. Por otra parte, debido a su tamaño, la pupila es menos probable de ser ocluida por los párpados. La principal desventaja de técnicas de imagen infrarroja es que no se puede utilizar al aire libre, durante el día, debido a la iluminación infrarroja ambiente.



**Fig.1.4** Detección del contorno exterior del iris (izda.) y de la pupila (dcha.)

El seguimiento de la mirada utilizando iluminación infrarroja permite dos técnicas para la detección de la pupila: la técnica de pupila brillante y la técnica de pupila oscura (ver fig. 1.6). También puede utilizarse la combinación de las dos técnicas para conseguir un sistema más preciso [14].

La técnica de pupila brillante ilumina el ojo con una fuente IR que se encuentre en o muy cerca del eje de la cámara. El resultado de la iluminación es tal que la pupila está claramente delimitada como una región brillante debido a la naturaleza de la parte reflectiva, la parte posterior, del ojo. La técnica de pupila oscura por el contrario ilumina el ojo con una fuente situada fuera del eje de la cámara siendo así la pupila la parte más oscura del ojo en la imagen. En ambos métodos, la reflexión especular de la primera superficie de la fuente de iluminación fuera de la córnea (el elemento exterior-óptico del ojo) también es visible. El vector entre el centro de la pupila y del reflejo de la córnea se suele utilizar en lugar del centro de la pupila como única indicación, para detectar la dirección de la mirada.



**Fig.1.5** Técnica de pupila oscura (izda.) y técnica de pupila brillante (dcha.)

La diferencia de posición entre el centro de la pupila y el reflejo producido por la córnea (primera reflexión de Purkinje) varía con la rotación del ojo pero con un pequeño movimiento de la cabeza puede mantenerse prácticamente constante. Por ello, una de las soluciones vigentes es las imágenes dual-Purkinje en las cuales se miden la primera y cuarta reflexión de Purkinje para separar las traslaciones y rotaciones de los movimientos del ojo. Sin embargo, esta solución sigue sin ser lo suficientemente precisa con lo que el usuario deberá utilizar indistintamente un reposa cabezas.

### 1.2.1 Configuraciones posibles

Todas estas técnicas anteriormente citadas son aplicables a cualquiera de las configuraciones de sistemas que se van a exponer a continuación.

En cuestiones de diseño, los sistemas VOG podemos agruparlos en dos: aquellos sistemas de eye-tracking montados en la cabeza (fig. 1.6) y los sistemas remotos, que serán detallados en el apartado 1.2.2. Los primeros son preferidos para aplicaciones que requieren grandes y rápidos movimientos de la cabeza; no es conveniente utilizarlos en aplicaciones que requieren una continua mirada de vigilancia durante largos períodos de tiempo (por ejemplo, el control del cursor para las personas con discapacidad motriz) o en aplicaciones que involucran a los niños. Para estas aplicaciones, los sistemas a distancia son los preferidos.



**Fig.1.6** Sistema de eye-tracking montado en la cabeza.

Recientemente, la invasión del seguimiento de la mirada ha sido reducida considerablemente con los avances en la miniaturización de los sistemas que son montados sobre la cabeza en los ojos-seguidores. Los sistemas remotos reducen la invasión respecto a los anteriores, sin embargo, pueden sufrir de una reducción de la precisión. Teniendo en cuenta estos avances, el obstáculo restante más importante es el costo. Los precios de estos sistemas no están puramente asociados con el hardware, como en el precio de una cámara digital de alta resolución, sino con un software personalizado, integrados a veces con especializados procesadores digitales para obtener rendimiento de alta velocidad.

Entre los sistemas remotos de eye-tracking, la configuración más simple consta de una sola cámara y una fuente de luz (la misma configuración que la figura 1.7a con un LED menos). No obstante, para calcular la dirección de la mirada con alta precisión no es suficiente el uso exclusivo de este sistema. Para satisfacer esta limitación habrá que mantener la cabeza fija en relación con el sistema y habrá que estimar la distancia entre el ojo y la cámara. Por lo general

estos sistemas suelen calcular la dirección de la mirada utilizando el vector del centro de la pupila al centro de la reflexión de la córnea con respecto al eje de la cámara o bien disponer de medios para estimar la posición del ojo en el espacio con ayuda de espejos por ejemplo.

La segunda configuración (ver fig.1.7a) que puede presentar un sistema remoto es la configuración más simple que permite la estimación del *PoR* a partir de los centros de la pupila y reflejos que no requiere de un cálculo adicional. Se trata de la configuración de partida que será analizada en este proyecto, que cuenta con una cámara y dos iluminadores una a cada lado de la pantalla. Aunque esta configuración no implique ninguna restricción en cuanto a movimientos de la cabeza y no requiera la utilización de ningún dispositivo adicional para estimar la posición del ojo en el espacio, para conseguir una precisión del sistema óptima, una de las dos soluciones será necesaria.

El resto de los sistemas que se van a presentar cuentan con más de una cámara para estimar la posición del ojo en 3D. El primero cuenta con dos cámaras estéreo (ver fig.1.7c). Esta configuración tiene dos algoritmos para acomodar los cambios de posición de los ojos:

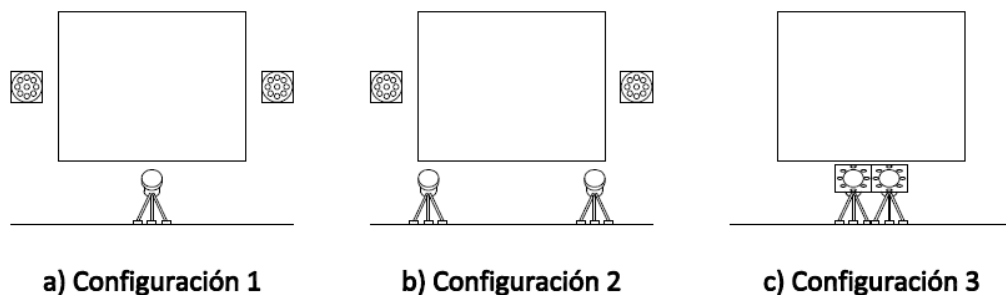
1.- Basándonos en la posición del globo ocular obtenido, la dirección de la mirada calculada puede ser actualizada ante cambios de posición de la cabeza por una propuesta dinámica del modelo computacional de posición de la cabeza de compensación.

2.- A partir del seguimiento de la mirada en 3D propuesto, podemos estimar los ejes virtuales en 3D directamente sin funciones de asignación.

Ambos métodos propuestos pueden alcanzar la precisión de menos de 1 grado a una distancia de unos 60cm a la pantalla.

Este sistema contiene los iluminadores en el eje de la cámara por lo que para la detección de la pupila se utilizará la técnica de pupila brillante.

El siguiente sistema cuenta con varias cámaras y dos iluminadores y se denomina sistema multi-cámara (ver fig. 1.7b), cada una de estas puede estimar la mirada de forma independiente, por lo tanto, permite grandes movimientos de la cabeza. La precisión de este sistema es también de menos de 1 grado, permaneciendo a una distancia de unos 60cm del sistema. Esta configuración contiene los iluminadores fuera del eje de la cámara por lo que para la detección de la pupila se utilizará la técnica de pupila oscura.



**Fig. 1.7** Esquema de tres de las posibles configuraciones

### 1.2.2 Sistemas basados en modelos matemáticos y sistemas basados en ecuaciones polinómicas.

Los sistemas de eye-tracking pueden ser clasificados en dos aproximaciones: los basados en modelos matemáticos o los basados en métodos polinómicos.

Los primeros exploran la geometría subyacente, como la posición de la cámara, tamaño de la pantalla, las características del sujeto, y sus conexiones geométricas, a fin de construir las expresiones matemáticas que describen la mirada en función de parámetros del sistema. Permiten la evaluación de la exactitud, de la sensibilidad con respecto a los parámetros específicos, y del número de puntos necesarios para la calibración. Hay dos inconvenientes fundamentales de estos modelos. En primer lugar, son más difíciles de construir. En segundo lugar, a excepción de unos pocos casos, como el método propuesto por Yoo y Chung [5], requieren una calibración adicional de parte o toda la geometría del sistema, es decir, pantalla, iluminación y cámara de [6] [9].

Los métodos polinómicos son generalmente expresiones polinómicas que describen el punto donde se fija la mirada ( $PoR$ ) como una función de las features de la imagen y un conjunto de coeficientes desconocidos que se deducen durante el proceso de calibración del sujeto. Los coeficientes se deducen por medio de un proceso de adaptación numérica tal como la regresión lineal múltiple. En ellas se describen el Punto donde se fija la mirada ( $POR$ ), es decir, el punto que se contempla en la pantalla ( $PoRX$ ,  $PoRY$ ), como una expresión polinómica utilizando estas características de la imagen seleccionadas como entrada. Comparado con los sistemas basados en modelos matemáticos, los métodos de asignación aportan mucha menos información sobre el comportamiento intrínseco del sistema, sin embargo son mucho más sencillos de construir y no requieren hardware adicional para la calibración. Ramanauskas ha demostrado que en términos de precisión los enfoques basados en modelos matemáticos o en ecuaciones polinómicas pueden ser muy similares [22]. Sin embargo, la tolerancia ante desplazamientos de la cabeza de las aproximaciones mediante ecuaciones polinómicas como las de modelos matemáticos dependen directamente del sistema utilizado es decir, de la cámara y la iluminación. Ante desplazamientos del usuario la detección de las features puede empeorar debido al cambio de iluminación por la posición de los iluminadores original y el desenfoque para ambas aproximaciones.

Hay tres aspectos fundamentales para diseñar una expresión polinómica: las características o features de la imagen de para utilizarlas como input de la expresión de estimación, el grado del polinomio, y el número de términos. Estos aspectos determinan las características esenciales del sistema, tales como el número de puntos de calibración necesarios (directamente proporcional al tiempo de calibración), la precisión y el rango tolerable de movimientos de la cabeza.

En este proyecto se va a utilizar un sistema basado en ecuaciones polinómicas a partir de la detección de las features del ojo relacionadas con la dirección de la mirada.

#### Estudios relacionados

La Doctora Arantxa Villanueva [7] plantea el desarrollo de diferentes modelos matemáticos que establecen conexión entre alguna de las características extraíbles de las imágenes captadas por el sistema VOG y el movimiento del ojo. Estos modelos matemáticos desarrollados y basados en parámetros geométrico físicos presentan ventajas como un control

sobre el comportamiento del sistema, su estabilidad, precisión, sensibilidad, fuentes de error, etc.

Los modelos matemáticos desarrollados en la esta tesis pueden clasificarse en tres grupos atendiendo al tipo de características de la imagen en las que se basa:

- **Modelos basados en puntos:** centro de la pupila y reflejos (1ª imagen Purkinje).
- **Modelos basados en formas:** contorno del iris.
- **Modelos híbridos:** una combinación de los dos anteriores.

Debido a la reflexión corneal ha sido difícil de conseguir una expresión analítica adecuada. Este motivo provoca que el único método (entre los propuestos) válido sea un modelo híbrido basado en el contorno de la pupila y varios reflejos. Este modelo final constituye un modelo matemático para el seguimiento de la mirada basado únicamente en parámetros físicos, al tiempo que emplea un hardware mínimo.

David Beymer y Myron Flickner [4] en el artículo en el cual proponen un sistema de seguimiento de la mirada utilizando una cabeza activa estéreo, proponen una aproximación que modela la anatomía de un ojo en 3D. Este modelo incluye el globo de la córnea, la pupila, y el offset angular de la fovea desde el eje óptico para reconstruir el eje visual. El sistema Fregase [13] también propone un modelo en 3D para el seguimiento del ojo, pero, solo modela la córnea como una esfera, mientras que el modelo anterior explora también con una elipse y la fovea no está expresamente modelada. Utilizando este modelo, se realizó un seguimiento del ojo utilizando una alineación aproximada. Para una posición hipotética, la pupila del ojo del modelo puede ser refractada a lo largo de la córnea y proyectada en la imagen. Estos rasgos del modelo proyectados pueden ser relacionados con los rasgos extraídos de la imagen. Este sistema permite movilidad de la cabeza y tiene una precisión de 0.6 grados considerando la distancia relativa entre el usuario y la cámara de 60cm.

D. Li, D. Winfield y D. J. Parkhurst [2] desarrollan un algoritmo híbrido que combina aproximaciones para el seguimiento del ojo basados en modelos y en features. Tanto la reflexión de la córnea como la pupila son localizadas mediante técnicas adaptables basadas en features. Luego se aplica el paradigma RANSAC para maximizar la precisión del ajuste de la elipse en la presencia del error cometido por la detección de las features. Finalmente es cuando se aplica una aproximación basada en un modelo matemático para refinar aún más el ajuste. Este algoritmo basado en modelos y en features consigue más precisión que los basados en features únicamente y consume menos tiempo que los basados en modelos.

Centrándonos en los sistemas polinómicos en los que se centra el presente proyecto, analizaremos estudios anteriores relevantes para nuestro trabajo. Como se ha señalado anteriormente, existen varios criterios de diseño a la hora de modelar o construir un sistema video-oculográfico basado en ecuaciones polinomiales tales como el orden de la expresión matemática, parámetros o variables predictivas, rango de funcionamiento, etc. Tradicionalmente, existen una serie de ideas preconcebidas a tal respecto. En este sentido, Morimoto et al. utiliza una aproximación basada en ecuaciones polinómicas y propone la minimización del error al aumentar el orden del polinomio, pero sólo en condiciones estacionarias [19]. Es decir, establece un vínculo directo entre el orden del polinomio y la minimización del error. Por otro lado, los resultados de White et al. [23] sugieren que, con respecto al movimiento de la cabeza, la exactitud con las expresiones de orden superior eran un poco mejores que las obtenidas con modelos más simples. Brolly et al. [24] experimentó con polinomios de hasta tercer orden y obtuvo una ligera mejora más respecto a las

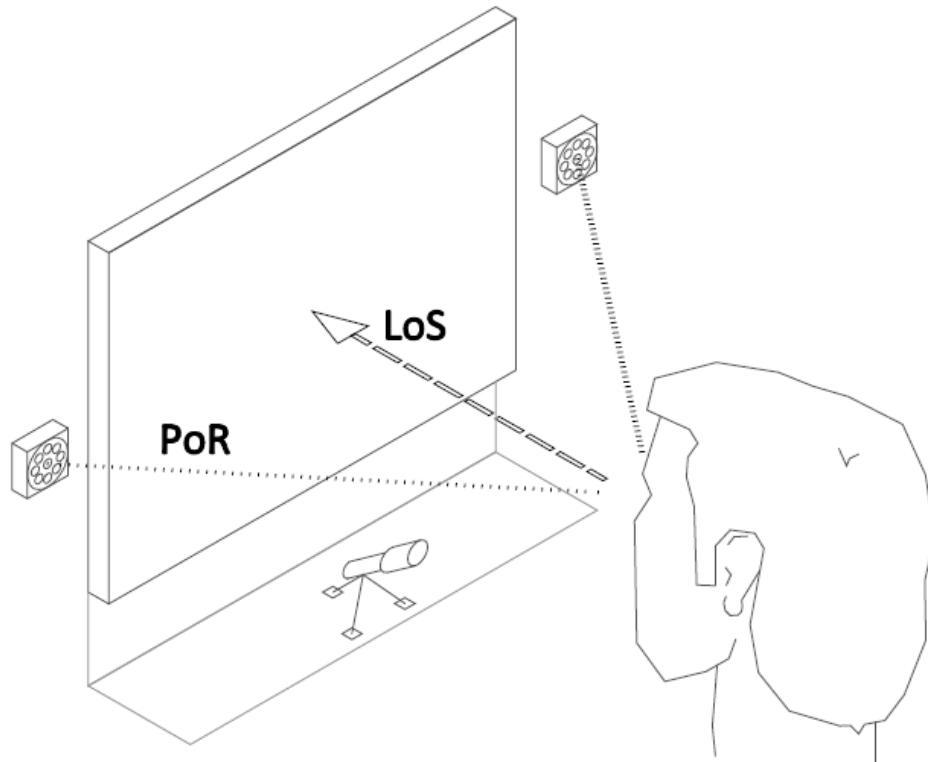


ecuaciones de segundo orden. Sin embargo, debido a la metodología experimental empleada por White et al. y el sistema de tres cámaras utilizadas por Brolly et al. no está claro si las conclusiones son aplicables a otros sistemas VOG. Recientemente Cerrolaza et al. [1] concluye que el efecto de aumentar el orden y número de términos de la ecuación de calibración no mejora el comportamiento ni la precisión del sistema. Este resultado es de gran importancia para el proceso de optimización del tiempo de calibración ya que demuestra la existencia de modelos simples y robustos que requieren menos puntos de calibración.



## 2 Descripción del sistema de eye-tracking utilizado

El sistema eye-tracker que se ha utilizado para la realización de este proyecto, como se ha comentado en el apartado 1.2, es un sistema basado en video-oculografía. Concretamente esta formado por dos fuentes de iluminación, LEDs infrarrojos, posicionados en los laterales de la pantalla a mitad de altura de la misma y una cámara de vídeo delante de la pantalla en el centro orientado hacia el ojo del usuario.



**Fig. 2.1** Configuración del sistema de partida utilizado

La luz infrarroja emitida por los LEDs es imperceptible por el ojo humano evitando distracciones del usuario, pero su espectro está centrado en 890nm, es decir, dentro del rango de actuación de la cámara. Además, esta luz no es nociva para los usuarios.

El objetivo del uso de este sistema es calcular el punto donde se fija la mirada o el *PoR*, que será considerado como la intersección entre el eje visual y la propia pantalla.

El primer paso del análisis será la obtención de la imagen a través del sistema de video-oculografía. Utilizando algoritmos de procesamiento de imagen se extraen de las imágenes las características del ojo que se asocian al movimiento del mismo. En este caso calculamos la posición del centro de la pupila y del primer reflejo de la córnea producido por cada uno de los LEDs (primera imagen de Purkinje).



Fig. 2.2 Ejemplo de imagen capturada por el sistema

## 2.1 Calibración

Para iniciar el seguimiento de la mirada es necesaria antes una sesión de calibración debido a que los coeficientes de la ecuación del cálculo del *PoR* se verán modificados en cada sesión dependiendo de la posición de la cámara, del usuario y de la pantalla, y los LEDs.

La ecuación de calibración viene en forma de coeficientes indeterminados, como una expresión polinómica cuyas variables son combinaciones lineales de las características de la imagen, el centro de la pupila y de los reflejos, multiplicados por unos coeficientes a calcular.

$$P_0 = (C_0)^T \cdot Fe \quad (2.1)$$

$P_0$ : *PoR*, punto donde miras

$C_0$ : vector de coeficientes desconocidos a calcular en la calibración

$Fe$ : vector de características de la imagen y posibles combinaciones lineales de las mismas.

La sesión de calibración consiste en que el usuario observe sucesivamente una serie de puntos (una cuadrícula generalmente) calculando las posiciones del centro de la pupila y los reflejos al mirar cada una de las fijaciones, conocidas por el sistema.

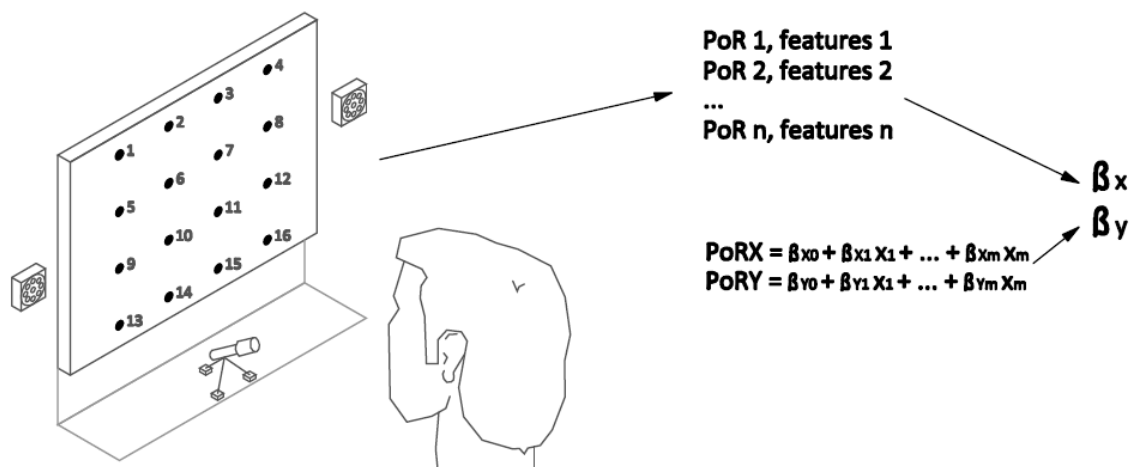


Fig. 2.3 Esquema del grid de calibración

De cada una de las coordenadas a las que el ojo mira se obtienen treinta imágenes o **capturas** y se observan un número de coordenadas  $(x_i, y_i)$  o **fijaciones** que depende del número de coeficientes de la ecuación del cálculo del *PoR*. De cada una de las treinta imágenes se escoge el 60% mejor de las mismas por mahalanobis, y mediante la regresión lineal se despejan los coeficientes específicos de cada usuario y sesión.

El cálculo del *PoR* se realiza mediante dos ecuaciones, una para el cálculo de la coordenada  $X$  y otra para la coordenada  $Y$ .

### 2.1.1 Análisis de regresión lineal

El análisis de regresión es una herramienta estadística que se basa en la relación existente entre dos o más variables cuantitativas de tal modo que una de ellas pueda ser deducida a partir del resto. Dicha variable es la denominada variable respuesta. El resto de variables reciben al mismo tiempo el nombre de variables predictoras.

Mediante esta herramienta estadística podremos conseguir un modelo predictivo capaz de estimar con la mayor fiabilidad y precisión posibles futuros comportamientos del sistema aunque el modelo no sea realista o no sea capaz de modelar la naturaleza del sistema.

El modelo de regresión lineal que vamos a utilizar es el modelo de **regresión lineal múltiple** en donde la variable dependiente se define en función de una constante  $C_0$  y varias variables independientes, parámetros que deben ser estimados, a partir de varias muestras de datos.

El método empleado para la determinación de estas variables independientes es el método por mínimos cuadrados ordinarios (MCO, o del inglés OLS) el cual nos permite obtener aquellos coeficientes que minimizan la suma del cuadrado de los errores en la estimación.

La ecuación del modelo podría escribirse como sigue:

$$Y_i = C_0 + C_1 x_{i,1} + C_2 x_{i,2} + C_3 x_{i,3} + \dots + C_p x_{i,p} + e_i \quad (2.2)$$

$Y_i$  es el valor de la variable respuesta o variable dependiente en la muestra  $i$ -ésima.

$x_{i,k}$  valor de la observación  $i$ -ésima de la variable independiente  $k$ -ésima.

$C_0$  constante de la regresión.

$C_k$  coeficiente de la variable independiente  $k$ -ésima.

$p$  número total de predictores.

$e_i$  término de error.

El término de error es desconocido y se le supone normalidad de media 0 y varianza  $C_0$ . Una vez que el modelo ha sido estimado mediante el procedimiento de los mínimos cuadrados podemos definir de forma alternativa los residuales de la regresión como:

$$e_i = \bar{Y}_i - Y_i \quad (2.3)$$

En nuestro caso para cada una de las ecuaciones de calibración utilizadas para cada coordenada ( $X$  o  $Y$ ) se realizará una regresión lineal múltiple independiente en cada calibración. La variable dependiente corresponde con el *PoR* conocido por el sistema y las muestras de datos serán las variables relacionadas con la posición central de la pupila y los reflejos del correspondiente *PoR* real. Estas variables más tarde serán definidas al indicar la ecuación de calibración de partida que será utilizada. Y los términos independientes a determinar son los coeficientes que multiplican cada una de las variables o muestras de datos.

La ecuación anteriormente definida (2.2) será la que utilizemos para calcular el error que cometemos con la ecuación de calibración para la estimación del  $PoR$  (frente al  $PoR$  real).

## 2.2 Inconvenientes que presenta el sistema

Este tipo de sistemas basados en ecuaciones polinómicas son más sencillos de construir que aquellos basados en modelos matemáticos. Sin embargo, los sistemas basados en modelos matemáticos presentan una precisión mayor que los basados en ecuaciones polinómicas ante desplazamientos de los usuarios.

Además, una de las limitaciones de los sistemas video-oculográficos que utilizan una sola cámara es la imposibilidad de determinar la localización en el espacio 3D de los objetos; imposible sin un sistema calibrado y el sistema de partida que utilizaremos no podrá medir la distancia del usuario a la cámara.

El error cometido por el sistema ante movimientos del usuario principalmente se debe a los desplazamientos realizados en el eje cámara usuario (eje  $Z$ ). Este eje es el más crítico ya que es el que mayor error presenta y en el que el usuario al interactuar con el ordenador más movimientos realiza. Además, la posición del ojo a través de una sola cámara la podemos calcular en 2D (eje horizontal y vertical) pero, es más difícil de calcular la distancia relativa entre el usuario y la cámara.

H. Morimoto y Marcio R.M. Mimica [19] realizan un estudio del error cometido al estimar el  $PoR$  ante desplazamientos de usuarios. Primero, evalúan su sistema frente a un movimiento en el eje horizontal (eje  $X$ ) respecto a una posición inicial y más tarde seguirán el mismo procedimiento con un movimiento en el eje  $Z$ . En el primer caso, concluyen que el movimiento en el eje  $X$  no afecta a la estimación del  $PoR$  sin embargo, el del eje  $Z$  aumenta el error, es alrededor de 3 veces mayor.

Jixu Chen, Yan Tong, Wayne Gray, y Qiang Jiz [9] realizan un estudio para analizar el ruido en 3D causado por el modelo matemático presentado en el que concluyen que la principal influencia de este ruido es debido principalmente al eje  $Z$ , lo cual causará el ruido de la estimación de la mirada. Para minimizar este error ellos proponen una restricción en la distancia virtual entre la pupila y la córnea de su modelo matemático en el eje  $Z$  manteniendo intactos el eje vertical y horizontal.

Varios sistemas [16], [17], [18], [19], [20], [21], los cuales han sido contruidos basándose en la técnica PCCR (centro de la pupila – centro de los reflejos) muestran que si el usuario tiene la capacidad de mantener la cabeza fija, o a través de la ayuda de un reposa cabezas para restringir los movimientos de la cabeza, se puede lograr una alta precisión de resultados del seguimiento de la mirada. En concreto, el error promedio puede ser menor de  $1^\circ$  ángulo visual, que corresponde a menos de 10mm en la pantalla del ordenador cuando el sujeto está sentado a 550mm desde la pantalla del ordenador. Pero a medida que la cabeza se mueve alejándose de la posición original en el que el usuario realiza la calibración, la exactitud de estos sistemas de seguimiento de la mirada decae dramáticamente.

Sin embargo, los sistemas basados en modelos matemáticos [2], [4], [6], [9], son más precisos que los anteriores frente a desplazamientos de la cabeza y también en la posición de calibración. No obstante estos sistemas tienen diferentes configuraciones (excepto [2]), es decir utilizan más de una cámara para poder tomar imágenes en estéreo. El sistema utilizado por el segundo de ellos [4] se divide en dos: estéreo de gran angular para la detección de rostros y estéreo de banda estrecha para el seguimiento de los ojos. El modelo matemático presentado por [6] admite cualquier configuración posible desde una cámara y un reflejo, la

cual requiere de información adicional de la posición del ojo; una cámara y dos reflejos, en este caso no hará falta información adicional pero el usuario deberá estar quieto; y utilizando más cámaras para averiguar la posición en 3D del ojo el usuario podrá desplazarse sin pérdidas en la precisión del sistema. El último sistema [9] también se compone de dos cámaras en estéreo para realizar el tracking y dos iluminadores IR sobre cada una de las cámaras permitiendo también hallar la posición de los rasgos del ojo en 3D.

### 3. Optimización del proceso de calibración de los sistemas video-oculográficos

#### 3.1 Ecuación de calibración de partida

En el estudio taxonómico de la regresión polinomial aplicada a la calibración de un sistema video-oculográfico de J.J. Cerrolaza, A. Villanueva y R. Cabeza [1] se comprueba la robustez y la exactitud de diferentes ecuaciones de calibración con el objetivo de encontrar una ecuación óptima que minimice el error ante desplazamientos de la cabeza. En este estudio se han considerado cuatro arquitecturas de sistema diferentes: centro de la pupila y una fuente IR (I), centro de la pupila y dos fuentes IR (II), la elipse de la pupila con una fuente IR (III) y elipse de la pupila con dos fuentes IR (IV). En función de la arquitectura utilizada se forman distintos grupos de variables.

Las variables básicas utilizadas en las distintas ecuaciones de calibración obtenidas a través del procesamiento de imagen son: el centro de la pupila y el centro de los reflejos ( $C$ ,  $R1$  y  $R2$ ) y las variables elípticas, los semiejes del iris ( $a, b$ ) a partir de los cuales se calcula la excentricidad. Además de estas variables se derivan unas nuevas, que mejoran significativamente la robustez del sistema (fig. 3.1): la distancia entre los reflejos, ( $D = |R1 - R2|$ ), los vectores pupila reflejos ( $V1 = C - R1$  y  $V2 = C - R2$ ), los vectores pupila reflejos normalizados respecto a la distancia entre los reflejos ( $VN1 = (C - R1)/D$  y  $VN2 = (C - R2)/D$ ) y los vectores medios ( $V = (V1 + V2)/2$  y  $VN = (VN1 + VN2)/2$ ).

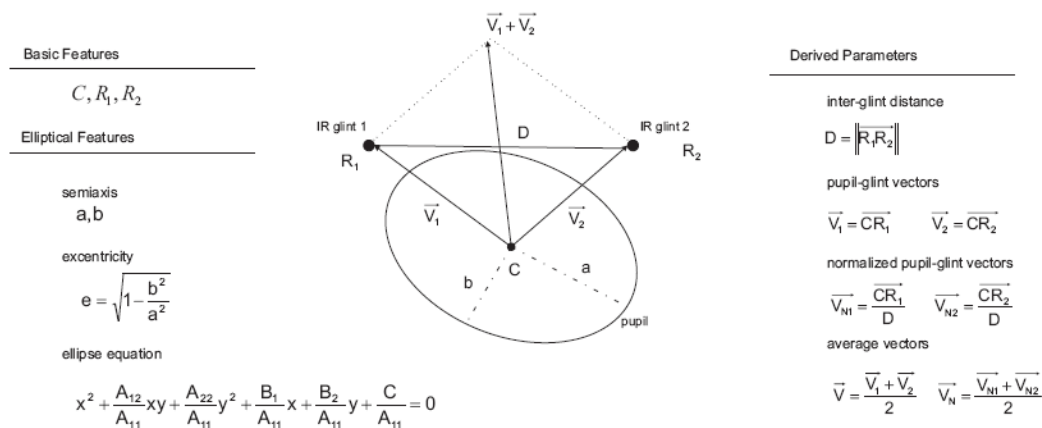


Fig. 3.1 Características de mapeo

En la realización de este estudio participaron 11 usuarios en el proceso de obtención de datos. Cada uno de ellos fue situado a 63cm de la cámara y se le pidió mirar una secuencia de puntos que forman una cuadrícula de calibración de  $4 \times 4$ . Estos datos serán utilizados para calcular los coeficientes de la ecuación de calibración cuya exactitud fue probada utilizando rejillas de  $8 \times 8$ , a 63cm de la cámara y a 5cm hacia delante y hacia atrás de esta posición. Solo se implementaron movimientos en el eje cámara nariz porque se consideran los más críticos. Además, el rango de estos movimientos es limitado ya que para este estudio se ha utilizado un objetivo de la cámara de 35mm el cual limita el rango de los movimientos del usuario porque sino el ojo saldría del plano imagen debido a la limitación del campo de visión. Todas las pruebas se realizaron sobre un reposa cabezas.

El criterio seguido para la elección de ecuaciones ha sido la intersección de las ecuaciones con el menor error medio, ecuaciones con el error máximo más bajo y las ecuaciones de menor desviación estándar de ambos errores.

Obteniendo un resultado similar con muchas de las ecuaciones estudiadas, finalmente se escoge una función cuyo requerimiento de calibración sea mínimo. Es decir, una expresión con un número de variables reducido. Logrando así, optimizar el proceso de calibración, tanto en el tiempo como en número de puntos.

$$PoRX = Cx_0 + Cx_1 V_N x + Cx_2 V_N x^3 + Cx_3 V_N y^2 \quad (3.1)$$

$$PoRY = Cy_0 + Cy_1 V_N x + Cy_2 V_N y + Cy_3 V_N x^2 V_N y - \Delta y(D) \quad (3.2)$$

La estimación del *PoR* se realiza mediante dos ecuaciones una para cada una de las coordenadas de la pantalla. Además como novedad en este estudio se consideran diferentes ecuaciones para cada coordenada rompiendo otra de las convenciones existentes entre los sistemas VOG basados en ecuaciones polinómicas.

### 3.2 Error que presenta el sistema con el movimiento de la cabeza

Los eye-trackers basados en video-oculografía registran el movimiento ocular desde la distancia, ubicados delante del monitor resultando ser poco invasivos y por tanto generan menos ruido en procesos de evaluación. No obstante, frente a los eye-trackers que se colocan en la cabeza del participante presentan una desventaja importante, ya que estos son adecuados para que el usuario tenga total libertad de movimiento de la cabeza.

Sin embargo, la ecuación de calibración utilizada, considerada anteriormente, comete error con movimientos de la cabeza del usuario. Con el objetivo de aumentar la robustez del sistema ante desplazamientos del participante se ha propuesto este proyecto. Debido a la imposibilidad de la localización del ojo en 3D mediante el sistema propuesto, puesto que no dispone información espacial, recurriremos a otra herramienta (FaceAPI) como posible solución. FaceAPI es un software de seguimiento de caras que permite calcular la posición de la cabeza en 3D.

Inicialmente, el objetivo final sería sincronizar los dos programas el de seguimiento de caras con el de seguimiento de la mirada corrigiendo la ecuación de calibración dotando al sistema de una mayor robustez frente a desplazamientos del usuario.

Para abordar este propósito se han estudiado los siguientes aspectos:

- Hallar la posición de la cabeza del usuario a través del uso de un software de seguimiento de caras: FaceAPI.
- Estudio de la cuantía del error presentado por el sistema ante desplazamientos de los usuarios controlados y al mismo tiempo cuantificados por FaceAPI.
- Sincronización de la aplicación FaceAPI con el resto del sistema de eye-tracking.

## 4 Shakti

El sistema Shakti es un software desarrollado por la Universidad Pública de Navarra que realiza el seguimiento de la mirada permitiendo el uso de ordenador mediante el ojo, moviendo el cursor con la mirada **a tiempo real**. Shakti ha sido desarrollado para facilitar la comunicación entre personas discapacitadas permitiéndoles el control total del ordenador mediante el movimiento del ojo utilizando el sistema VOG antes descrito para colocar el puntero del ratón en la pantalla.

Una vez el usuario se encuentre en el lugar de la pantalla deseado, puede realizar “click” del ratón, bien parpadeando o bien por latencia.

El hardware que utilizaremos, como ya hemos citado anteriormente, cuenta con una cámara (PointGray) y dos iluminadores IR a ambos lados de la pantalla. Tanto la cámara como dichos iluminadores serán conectados a una fuente de alimentación. La cámara se sitúa sobre una base delante de la pantalla y se conecta al puerto Firewire del ordenador. El objetivo que inicialmente se utiliza es de 25 o 35mm. Sin embargo, para realizar este proyecto no ha sido suficiente con el rango de movimientos permitido por este objetivo y se ha empleado uno de 16mm también.

Existen tres modos de funcionamiento de la aplicación PJ1: modo Usuario, Cuidador y Mantenimiento. En el primer modo el usuario puede habilitar/deshabilitar el control del cursor del ojo y realizar las calibraciones pertinentes. El modo Cuidador permite configurar la aplicación adaptándola a cada usuario.

El monitor de la pantalla utilizado es de 17” (337.9H x 270.33V mm) con una resolución de 1280 x 1024 píxeles. Es decir, 0.26398 mm/px.

La cámara utilizada con sensor de 1/3 (3.6 x 4.8 mm) tiene una resolución de la imagen de 1024 x 768 píxeles. Por lo que las dimensiones del píxel de la imagen serán de 7.4  $\mu$ m.

### 4.1 Tratamiento de los datos

Independientemente de que la ecuación de calibración de partida solo requiera de una grid de calibración de 8 puntos, para un estudio más completo y exacto en las pruebas realizadas con Shakti en este proyecto se van a utilizar grids de 16 puntos.

En cada calibración concretamente tendremos 16 fijaciones distintas y para cada una de las fijaciones se obtienen cerca de 30 imágenes. De cada una de las 30 imágenes por fijación Shakti por procesamiento de imagen determina la posición del centro de la pupila y los dos reflejos, dos coordenadas por cada una de los puntos.

La información de Shakti es extraída a un archivo de texto en el que se incluye el tiempo de la captura de cada imagen, las coordenadas X e Y de cada una de las fijaciones conocidas por Shakti y la información correspondiente del centro de la pupila y los reflejos.

La información del tiempo viene dada en segundos con una precisión de microsegundos. Y la información de la localización de cada una de las features del ojo, el centro de la pupila y los dos reflejos viene dada en píxeles con una resolución de hasta 4 decimales.

#### 4.1.1 Filtrado de los datos

A pesar del uso de un reposa cabezas, es razonable suponer que no todas estas imágenes son igualmente adecuadas. Algunas son menos útiles, como las imágenes adquiridos durante



el parpadeo y, en consecuencia, algún tipo de filtrado de datos es necesario. Dado que no existe un subconjunto de características comunes a todas las ecuaciones de mapeo, se van a escoger criterios específicos de filtrado.

Cada imagen procesada puede ser considerada como un vector n-dimensional. Donde  $X_{ik}$  será el vector que contendrá n características de la imagen del usuario i-ésimo de la imagen k-ésima.

$$X_{ik} = \{x_{ik1}, x_{ik2}, \dots, x_{ikn}\}^T \quad (3.1)$$

donde  $x_{ikz}$  es la característica de mapeo z-ésima de la k = ésima imagen adquirida para cada punto. La media, y la matriz n × n de covarianza,  $S_i$ , se calculan de la siguiente manera:

$$\bar{X}_i = \frac{1}{30} \sum_{k=1}^{30} X_{ik} \quad (3.2)$$

$$S_i = \frac{1}{30} \sum_{k=1}^{30} (X_{ik} - \bar{X}_i)(X_{ik} - \bar{X}_i)^T \quad (3.3)$$

La distancia de Mahalanobis es una medida típica de la distancia entre las variables en un espacio multidimensional que se calcula así:

$$D_{ik} = \sqrt{(X_{ik} - \bar{X}_i)^T S_i^{-1} (X_{ik} - \bar{X}_i)} \quad (3.4)$$

Esta distancia nos ha permitido ordenar los n vectores en función de su distancia a la media. De esta manera, calculamos la distancia a la media de cada una de las imágenes de una fijación valorando cada uno de los elementos del vector  $X_{ik}$ , es decir, teniendo en cuenta la información de las coordenadas X e Y del centro de la pupila y de los reflejos. Esta distancia será el criterio seguido para el filtrado de las imágenes, en concreto se van a escoger únicamente el 60% de las imágenes más cercanas a la media.

#### 4.1.2 Cálculo de los coeficientes de la ecuación de calibración

Para calcular los coeficientes de calibración primero calcularemos los *vectores centro de pupila – reflejos de la córnea normalizados* ( $V_{Nx}$  y  $V_{Ny}$ ) a partir de los cuales calcularemos todas las variables utilizadas en la ecuación de calibración de la que partimos (ecuaciones 3.1 y 3.2).

En total tendremos 5 variables distintas calculadas por cada una de las imágenes. Estas variables serán los elementos característicos de la imagen que se utilizarán para realizar el filtrado. Es decir, los vectores  $X_{ik}$  estarán compuestos por 5 elementos: las variables de la ecuación calculadas a partir de la información de la pupila y los reflejos de cada una de ellas.

El filtrado de las imágenes también podría haberse realizado utilizando la información capturada por Shakti (pupila y reflejos) como componente de filtrado. No obstante ha sido comprobado que la diferencia entre ambas posibilidades no es significativa.

Después de realizar el filtrado procederemos al cálculo de los coeficientes utilizando solamente las variables escogidas por mahalanobis. El cálculo de los coeficientes se va a

realizar por regresión múltiple utilizando como respuestas el  $PoRX$  y  $PoRY$  real más las correspondientes variables como muestras de datos.

Para cada una de las ecuaciones se realizará una regresión independiente. En dicha regresión para calcular los coeficientes correspondientes a la ecuación en la coordenada  $X$  se utilizará como respuesta el  $PoRX$  real de cada fijación y las muestras de datos serán un vector de unos (para calcular el coeficiente independiente) y las variables consideradas a partir de las features del ojo. Para la otra ecuación se procederá indistintamente.

## 5 Descripción y caracterización de FaceAPI

Como ya ha sido comentado anteriormente, el sistema de eye-tracking que utilizamos en este proyecto no permite al usuario total libertad de movimientos manteniendo una buena precisión. Esto es, el error de la estimación del  $PoR$  mediante la ecuación de calibración utilizada aumenta ante estos movimientos. Con el objetivo de aumentar la robustez del sistema ante desplazamientos del participante, recurriremos a otra herramienta (FaceAPI), la cual permite localizar el ojo en 3D. El objetivo final es sincronizar los dos programas, el de seguimiento de caras con el de seguimiento de la mirada tratando de corregir la ecuación de calibración dotando al sistema de una mayor precisión frente a desplazamientos del usuario.

Para ello, en este apartado realizaremos un estudio previo de FaceAPI para conocer bien sus características y posibilidades.

FaceAPI es un software de seguimiento de caras que detecta la posición (en 3D) de la cabeza y la orientación de la misma, junto con la detección de la expresión facial. Fue desarrollado por Seeing Machines (<http://www.seeingmachines.com>). Esta es una empresa de dedicada a visión por computador que permite el rastreo de rostros humanos y rasgos faciales.

FaceAPI está disponible bajo tres tipos de licencia: no comercial, de desarrollo y de producción. En este caso contando con una licencia de desarrollo nos está permitido crear productos que utilizan el conjunto de características completo de FaceAPI. Esta licencia incluye un hardware USB que desbloquea la funcionalidad de FaceAPI. Para la distribución de FaceAPI hará falta una licencia de producción.

FaceAPI proporciona una serie de módulos de procesamiento de imagen creada específicamente para el seguimiento y la comprensión de las caras y rasgos faciales. Estos módulos de seguimiento son de gran alcance y proporcionan una fuente de información incorporable a otros servicios.

FaceAPI está estructurado como una biblioteca de archivos de cabecera en C, un archivo DLL de Windows, la documentación HTML, varias aplicaciones de ejemplo, una herramienta para la calibración de las lentes, etc.

El seguimiento de caras implementado por FaceAPI se lleva a cabo mediante una cámara independiente a la utilizada para el seguimiento de la mirada, funciona con cualquier cámara o webcam estándar. En este caso utilizaremos una webcam Logitech que permite grabar vídeos de hasta 2Mpíxeles de resolución y mantiene un frame rate adecuado, hasta 15 frames por segundo, para garantizar el funcionamiento de FaceAPI a tiempo real. El funcionamiento a tiempo real se considera a 30 frames por segundo.

FaceAPI además incorpora una aplicación, Smtrackfile, que permite calcular la localización de la cabeza y el seguimiento de rostros faciales de imágenes y de vídeos pregrabados. Admite archivos de película como avi, wmv, mov y otros formatos de archivo.

### 5.1 Funcionamiento de FaceAPI

FaceAPI realiza un seguimiento de la posición y orientación espacial de la cabeza en 3D (coordenadas  $X, Y, Z$ ) en relación a la posición de la cámara. También realiza una detección facial que discierne la posición de las features de la cara como las esquinas del ojo y las cejas, comisuras de la boca, y la nariz. En concreto, realiza un seguimiento de las cejas obteniendo 3 puntos de cada una, las 4 esquinas de cada ojo más el centro de la pupila, 6 puntos del contorno de la nariz y alrededor de los labios obtiene 8 puntos.

Al ejecutar FaceAPI, primero en la ventana consola aparece información a cerca de la versión de FaceAPI, de la licencia, de las cámaras detectadas y de los formatos en los que

puede trabajar el software (las resoluciones y el frame rate de cada uno). Además incluye un listado con diferentes opciones de tracking que pueden manejarse desde el teclado.

Estas opciones incluyen la posibilidad de reanudar el tracking y activar o desactivar el seguimiento de las cejas, los labios, la posición de la cabeza, los puntos destacados de la cara, la máscara de detección de la cara, información del rendimiento, escribir en la ventana consola la información de la posición de los puntos más destacados de la cara y la visualización de los tres ejes espaciales sobre la cara.

Por otro lado, se inicia un display de video donde se muestran las imágenes obtenidas por la cámara con dibujos de los contornos del tracking. El seguimiento comienza de forma automática, con readquisición automática en caso de pérdida del mismo.

FaceAPI al iniciar el seguimiento de caras necesita tener un tamaño mínimo de 50 píxeles de separación entre las esquinas externas de los ojos, una rotación máxima de  $15^\circ$  en el eje  $X$  y en el eje  $Y$  y de  $30^\circ$  en el eje  $Z$ . Aunque una vez que ha iniciado el seguimiento a partir de ahí es capaz de mantenerlo con un tamaño de la cara de 50 píxeles y admite rotaciones superiores a las mencionadas. El tiempo de adquisición es de entre 0,3 a 3 segundos según las especificaciones del programa. A partir de la primera adquisición, trabajando con el formato por defecto, es decir, con una resolución de la imagen de  $640 \times 480$  píxeles la tasa de frames por segundo es de 30. Esta tasa de frames por segundo será la necesaria para trabajar a tiempo real, un requisito indispensable a la hora de sincronizar esta herramienta con Shakti.

## 5.2 Caracterización de FaceAPI

Antes de proceder a integrar la información espacial de la cabeza proporcionada por la herramienta FaceAPI en el sistema de eye-tracking para poder dotar al sistema de eye-tracking de mayor robustez ante los desplazamientos del usuario, resulta esencial caracterizar dicha herramienta en términos de precisión, resolución y velocidad.

FaceAPI es un sistema propietario por lo que no podemos acceder a su configuración interna por ello tendremos que hacer ingeniería inversa, es decir, tratar de averiguar el funcionamiento interno de FaceAPI mediante una serie de pruebas, analizar la información que calcula para realizar el seguimiento de caras e intentar extraerla para integrarla en Shakti.

Como ya hemos comentado en el subapartado anterior, FaceAPI calcula información relativa a la posición de la cabeza y rasgos faciales. La información de la posición de la cabeza será extraída a un archivo de texto en tiempo real, es decir, manteniendo la tasa de frames por segundo en la que el programa calcula dicha posición. La extracción se realizará a un archivo de texto con información del tiempo de cada captura.

Para ello ha sido modificado el código de FaceAPI y ha sido creada una clase que permite instaurar un archivo, incluir información en el mismo y cerrarlo. El almacenamiento de esta información no se activará de forma automática sino que será ofrecida como una opción adicional de teclado.

### 5.2.1 Análisis de los datos

Como se ha comentado anteriormente, FaceAPI calcula por un lado la posición de la cabeza y de cada ojo relativa a la cámara y la rotación de la misma, e información de los puntos más característicos de los rasgos faciales.

La información de la posición de la cabeza se presenta en tres vectores, cada uno de ellos contiene una de las coordenadas de la posición de la cabeza por cada imagen adquirida mediante la cámara. Directamente relacionada con esta variable, la información de la rotación de la cabeza también se presenta para cada uno de los ejes  $X, Y, Z$ . Los datos de la posición del ojo siguen el mismo esquema que los datos de la posición de la cabeza pero con información

relativa a cada ojo. La variable que determina la expresión facial incluye la posición de varios puntos de los contornos de los rasgos faciales.

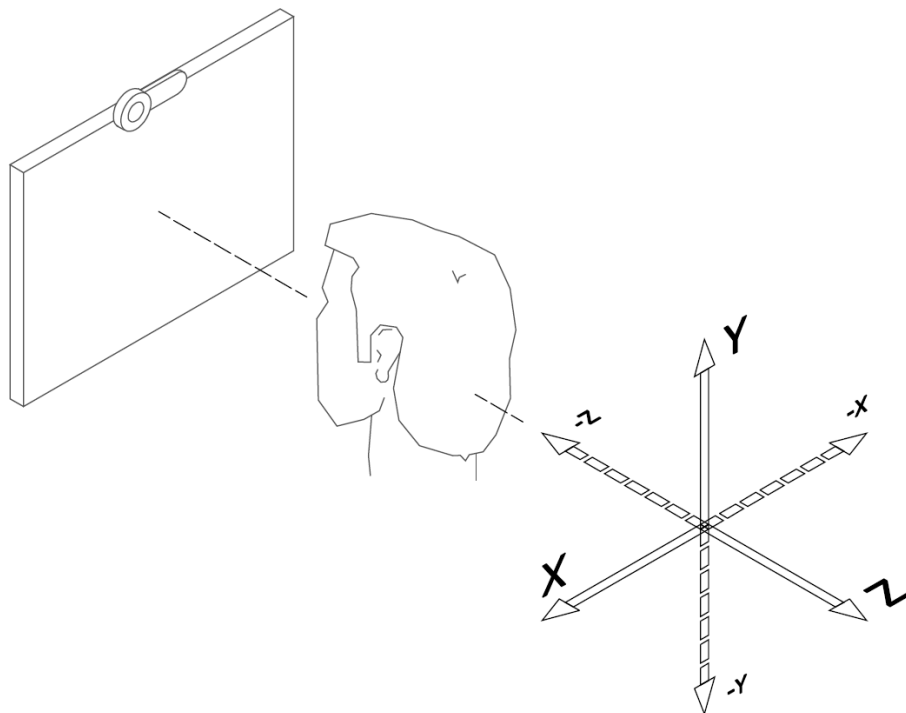
Sin perder de vista el objetivo del proyecto, la información que nos cuantifica los desplazamientos del usuario será la obtenida por la combinación entre la información de la posición y rotación de la cabeza.

### 5.2.1.1 Ejes

Tal y como hemos comentado anteriormente, para cumplir el objetivo de este proyecto, sólo será relevante la información de la posición y rotación espacial de la cabeza entre la que nos ofrece FaceAPI. Por tanto, antes de empezar a trabajar con estas variables se ha realizado un pequeño estudio a cerca del comportamiento de esta información.

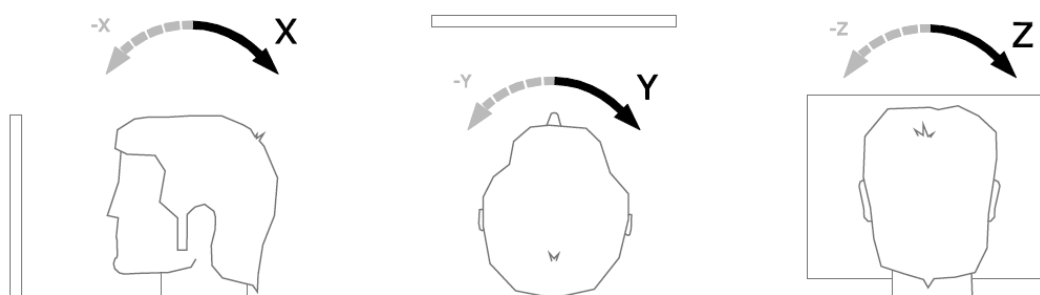
El primer paso será asociar las coordenadas  $X, Y, Z$  tanto de la posición como de la rotación de la cabeza con los ejes reales, realizando desplazamientos y asociándolos al cambio de los valores de alguna de las componentes de cada variable.

El eje  $X$  se considera el horizontal, el comprendido entre oreja a oreja bajo el supuesto de que el participante se encuentra posicionado en frente de la cámara. El eje  $Y$  se considera el eje vertical  $Y$  el eje cámara – nariz bajo la misma suposición, la distancia del usuario a la cámara. El sentido de los ejes puede verse en la figura 5.1.



**Fig. 5.1** Sentido de los ejes de traslación

Por tanto, la rotación en el eje  $X$  consiste en la rotación de la cabeza hacia arriba y hacia abajo, en el eje  $Y$  la rotación de la cabeza de izquierda a derecha y en el eje  $Z$  a la rotación de la cabeza de izquierda a derecha manteniendo siempre la nariz fija orientada hacia la cámara. El sentido de los ejes de rotación puede verse en la figura 5.2.



**Fig.5.2** Sentido de los ejes de rotación

#### 5.2.1.2 Magnitud y unidades de medida de la posición y rotación de la cabeza

Por otro lado, para seguir con la caracterización de FaceAPI será necesario conocer las unidades en las que la aplicación mide los desplazamientos y las rotaciones.

Para cuantificar las unidades de los desplazamientos y rotación se ha procedido de la misma manera que en el caso anterior, es decir, observando la evolución de las variables ante cambios del usuario provocados y controlados.

Primero hemos averiguado las unidades de la información relativa a la rotación de la cabeza. Desplazamientos en los tres ejes hacia ambos lados de unos  $45^\circ$  hemos comprobado que la rotación ha sido medida en grados. Esta prueba se ha realizado observando la evolución de la rotación al activar en la consola la opción de representación de los datos por pantalla. En cambio, al guardar en un archivo cada componente de la variable correspondiente a la rotación de la cabeza por separado esta información viene dada en radianes. A pesar de todo, en la memoria los datos guardados han sido pasados radianes a grados para un estudio más cómodo.

Procediendo de la misma manera, hemos comprobado que la posición de la cabeza de cada usuario la presenta en metros. Esta comprobación se ha realizado mediante desplazamientos controlados en los ejes  $X$  e  $Y$ . Sin embargo, la distancia entre el usuario y la cámara que indica la componente  $Z$  de la posición de la cabeza no se corresponde del todo con la realidad. Realizando el tracking de varios usuarios en la misma posición FaceAPI no ofrece el mismo resultado en  $Z$ . A partir de este resultado deducimos que FaceAPI puede que trabaje con un modelo matemático interno porque para obtener la localización espacial de un objeto a partir de un sistema sin calibrar de una única cámara, tal y como funciona FaceAPI, es necesario recurrir a un modelo matemático-estadístico previo. Es posible que a partir de una serie de parámetros de la imagen (medidas antropométricas como la distancia entre los ojos, la nariz, etc.) se realice algún tipo de calibración interna del modelo. De este modo, la distancia entre el usuario y la cámara corresponderá a aproximaciones realizadas a partir de dicho modelo.

Una vez estimadas las unidades de posición y rotación tendremos que comprobar la resolución de estas variables. En el caso de la información relativa tanto a la posición como a la rotación de la cabeza tenemos una resolución de cada variable de doble precisión (en C++), sin embargo a la hora de almacenar estas variables nos quedamos con una resolución de 5 decimales, la primera medida en metros y la segunda en radianes. Esta resolución sin embargo no indica que tengamos el mismo nivel de precisión de cálculo. El nivel de precisión será analizado en siguientes apartados.

### 5.2.1.2.1 Desplazamientos máximos

FaceAPI obviamente permite realizar unos desplazamientos limitados por el plano de imagen y la óptica de la cámara, y también está limitado por la rotación ya que nos da información de la posición de la cabeza a partir del seguimiento de caras realizado.

- Rotación máxima en el eje  $X$   $-30^\circ < X < 60^\circ$  Eje horizontal (de oreja a oreja)
- Rotación máxima en el eje  $Y$   $-90^\circ < Y < 90^\circ$  Eje vertical
- Rotación máxima en el eje  $Z$   $-90^\circ < Z < 90^\circ$  Eje cámara - nariz

Este software permite además una oclusión máxima del 50%.

### 5.2.1.3 Variación de los datos con la cabeza quieta (Ruido de FaceAPI)

Para completar este estudio hemos calculado el error cometido por FaceAPI en el cálculo de desplazamientos y rotaciones de la cabeza, y la variabilidad total del sistema causada por este error de FaceAPI más la variabilidad propia del usuario al intentar permanecer quieto.

#### 5.2.1.3.1 Con fotos de caras

Para comprobar el ruido cometido por el software, se han utilizado fotos de caras fijadas sobre un soporte y se ha guardado el seguimiento durante un periodo de un minuto y a continuación se ha calculado la desviación de los datos de cada variable.

Fotos de Caras	$\sigma$ posición (eje $X$ )	$\sigma$ posición (eje $Y$ )	$\sigma$ posición (eje $Z$ )
Foto 1	$4.81 \cdot 10^{-17}$	$7.82 \cdot 10^{-17}$	$4.00 \cdot 10^{-3}$
Foto 2	$3.95 \cdot 10^{-16}$	$1.58 \cdot 10^{-16}$	$6.33 \cdot 10^{-3}$
Foto 3	$1.46 \cdot 10^{-16}$	$1.21 \cdot 10^{-17}$	$9.71 \cdot 10^{-4}$

Tabla I. Desviación estándar de los datos de posición (en metros).

Fotos de Caras	$\sigma$ posición (eje $X$ )	$\sigma$ posición (eje $Y$ )	$\sigma$ posición (eje $Z$ )
Foto 1	$6.03 \cdot 10^{-2}$	$6.70 \cdot 10^{-2}$	$1.74 \cdot 10^{-2}$
Foto 2	$11.29 \cdot 10^{-2}$	$8.18 \cdot 10^{-2}$	$2.63 \cdot 10^{-2}$
Foto 3	$8.50 \cdot 10^{-2}$	$6.10 \cdot 10^{-2}$	$1.92 \cdot 10^{-2}$

Tabla II. Desviación estándar de los datos de rotación (en grados).

En las tablas anteriores observamos que los niveles de desviación de los datos son muy bajos, sobre todo en los datos relativos a la posición de la cabeza de los ejes  $X$  e  $Y$ , no tanto para el eje  $Z$ . La variabilidad de los primeros máxima es de 1cm y los datos relativos a la rotación en cambio varían entorno a  $1^\circ$  llegando incluso a los  $2^\circ$  en algunos casos.

### 5.2.1.3.2 Con usuarios reales

A la hora de realizar una calibración en Shakti, este considera que el usuario está quieto en una posición, sin embargo, a pesar de utilizar un chinrest para inmovilizarlo el usuario puede presentar movimientos involuntarios.

Para estudiar dicha variabilidad cuando el sistema es utilizado por usuarios reales inmovilizados por el reposa cabezas, realizaremos una prueba en la que calcularemos la posición y rotación de la cabeza de tres usuarios a partir de una misma posición.

Fotos de Caras	Posición (eje X)		Posición (eje Y)		Posición (eje Z)	
	Media (m)	$\sigma$	Media (m)	$\sigma$	Media (m)	$\sigma$
<b>Usuario 1</b>	-0.0321	$3.51 \cdot 10^{-4}$	-0.0096	$2.74 \cdot 10^{-4}$	0.7288	$2.7 \cdot 10^{-3}$
<b>Usuario 2</b>	-0.0321	$5.27 \cdot 10^{-4}$	-0.0207	$8.64 \cdot 10^{-4}$	0.8151	$7.7 \cdot 10^{-2}$
<b>Usuario 3</b>	-0.0378	$3.53 \cdot 10^{-4}$	-0.048	$9.7 \cdot 10^{-4}$	0.7552	$2.6 \cdot 10^{-3}$

**Tabla III.** Posición de la cabeza (en metros).

La variabilidad de los datos de cada usuario es muy baja y la variabilidad entre usuarios corresponde con los datos del error que FaceAPI asume, es decir 1cm. No obstante, teniendo en cuenta de que todas las pruebas están realizadas en la misma posición del reposa cabezas el error entre usuarios es alto en el caso de la posición en el eje Z. La diferencia entre usuarios en el eje Y puede ser debida a que FaceAPI no calcule altura a partir de la barbilla del usuario que es la posición de partida de todos ellos, sino de alguna otra característica de la cara la que esté a diferente distancia de la barbilla dependiendo del usuario. Sin embargo, la variabilidad de los datos en el eje Z es mayor: de casi de 10cm cuando en las especificaciones de FaceAPI se considera un error máximo de 1cm y al controlar la posición del usuario mediante un chinrest, la diferencia no puede ser tan alta.

Como se ha comentado anteriormente, el error cometido al calcular la posición en el eje Z entre los distintos usuarios puede ser consecuencia de la utilización de algún tipo de modelo matemático de fondo necesario ya que se está trabajando con solo una cámara. Se podría realizar un estudio en profundidad para establecer una relación entre la distancia Z y alguno de los parámetros antropométricos (ver apartado 5.4.2). Sin embargo, simplemente con estas pruebas ponemos en cuestión la utilidad potencial de esta herramienta para nuestros propósitos.

Fotos de Caras	Rotación (eje X)		Rotación (eje Y)		Rotación (eje Z)	
	Media (°)	$\sigma$	Media (°)	$\sigma$	Media (°)	$\sigma$
<b>Usuario 1</b>	-26.10	$9.1 \cdot 10^{-3}$	-5.28	$1.38 \cdot 10^{-2}$	2.35	$5.7 \cdot 10^{-3}$
<b>Usuario 2</b>	-21.51	$7.4 \cdot 10^{-3}$	-0.04	$1.28 \cdot 10^{-2}$	0.0327	$6.1 \cdot 10^{-3}$
<b>Usuario 3</b>	-9.58	$3.16 \cdot 10^{-2}$	-4.50	$2.0 \cdot 10^{-2}$	1.24	$8.3 \cdot 10^{-3}$

**Tabla IV.** Orientación de la cabeza (en grados).

La desviación de los datos de rotación de la cabeza también es baja entre usuarios pero, hay grandes diferencias en la rotación en el eje X entre usuarios. Sin embargo, esto pudiera ser provocado por la colocación en el chinrest de cada usuario.

En conclusión, al colocarse los tres usuarios en una misma posición de partida respecto a la cámara utilizada por FaceAPI, esta herramienta calcula su posición cometiendo un error de cerca de 1cm entre usuarios tal y como lo advierten las especificaciones de FaceAPI en el eje X



e  $Y$ , aunque la variabilidad de los datos de cada usuario es mucho menor. Sin embargo, la diferencia entre la posición calculada para cada usuario relativa al eje  $Z$  es mucho mayor, además, la variabilidad de los datos de este eje por usuario también lo es.

Al calcular la orientación de la cabeza también ocurre lo mismo que al calcular la posición del eje  $Z$  del usuario. Sin embargo, la rotación de la cabeza del usuario no es tan controlable por el chinrest como lo es la posición por lo que podría haber más diferencias entre usuarios.

### 5.3 Tiempos de captura

Para hacer posible la sincronía entre Shakti y FaceAPI será necesario el almacenamiento del tiempo de adquisición de cada uno de los seguimientos. En esta tarea han sido consideradas varias opciones como contadores. (La captura del tiempo se realizará de la misma manera en Shakti).

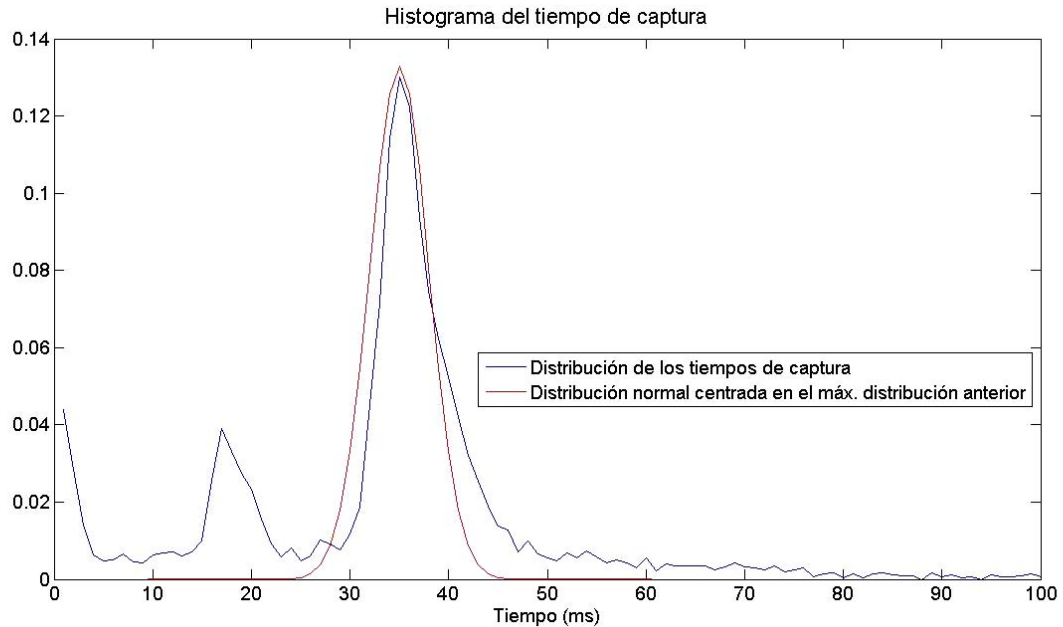
Las marcas de tiempo que se pueden obtener mediante contadores de Windows están limitadas a una resolución máxima de 10 o 15 milisegundos en función del hardware subyacente (estos contadores proporcionan la hora del sistema). Lógicamente para que la sincronización se produzca a tiempo real, es conveniente conseguir una mayor resolución. Se podría pensar que esto sería fácil con la API de `GetSystemTime` que devuelve una estructura que contiene un campo `SystemTime` en milisegundos documentado en MSDN como el almacenamiento de la milésima de segundo. A pesar de ello, con la simple prueba de recuperar la hora de sistema y sacarla por pantalla, se observa que la precisión es de 15ms aunque la resolución de la variable sea de milisegundos.

Para obtener la hora del sistema con mayor precisión la solución sería la utilización de contadores de rendimiento y sincronizar el valor de este contador al instante en el que la hora del sistema cambie. Sin embargo, esta manera de calcular la hora del sistema presenta unos problemas: la frecuencia reportada por `QueryPerformanceFrequency` y la falta de protección de los cambios del tiempo del sistema [37].

No obstante, ya que el único propósito del cálculo del tiempo es la sincronización de dos programas bastará con utilizar el contador de rendimiento `QueryPerformanceCounter` que nos devuelve el tiempo transcurrido desde el momento del encendido del ordenador sin necesidad de sincronizarlo con el tiempo del sistema. Pero, para mantener coherencia en ambos programas el tiempo será calculado acordemente. Esto será necesario ya que para poder sincronizar ambos sistemas se realizará una interpolación de los datos.

#### 5.3.1 Estudio de los tiempos de adquisición de FaceAPI

Una vez establecido el contador que utilizaremos para calcular el tiempo de captura continuaremos con la caracterización de FaceAPI. Esta vez aprovechando la información del tiempo hemos comprobado la tasa real de frames por segundo. En la gráfica de la figura 5.3 se muestra el histograma de la diferencia del tiempo de captura de datos de FaceAPI.



**Fig. 5.3** Distribución del frame-rate.

En la gráfica 5.3 vemos como el tiempo transcurrido entre capturas consecutivas se aproxima a una distribución normal con media de 35ms y varianza de 3ms. En vista de estos resultados observamos que para el funcionamiento a tiempo real la tasa de capturas por segundo es algo inferior a lo que debiese y esto perjudicaría la precisión del sistema al sincronizarlo con Shakti. Este problema será tratado y solucionado en el apartado 5.4.1.

## 5.4 Problemas en la adquisición de datos

A lo largo del estudio de FaceAPI nos hemos encontrado frente a dos problemas: el primero se muestra en el apartado inmediatamente anterior, debido a que el tiempo transcurrido entre capturas es elevado, supera los 30ms por lo que no estaría trabajando a tiempo real. Y el segundo problema es que FaceAPI no considera las mismas posiciones entre distintos usuarios como iguales.

### 5.4.1 Funcionamiento a tiempo real

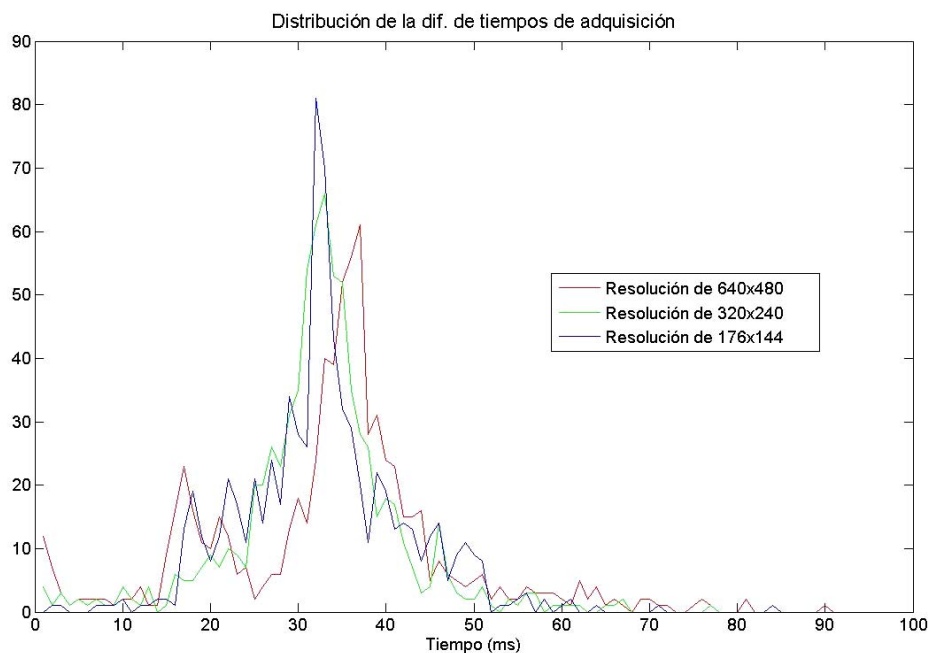
Partiendo de que el sistema no funciona a tiempo real, se sugieren dos soluciones para aumentar la tasa de frames por segundo o frecuencia de adquisición de datos.

La primera es efectuar otra modificación en el código para ofrecer una nueva opción ejecutable por teclado que consiste en visualizar o no el display de video con el dibujo del seguimiento ya que resulta innecesario en esta labor y supone un elevado coste computacional. Sin embargo, esta opción no origina resultados evidentes en la tasa de frames por segundo por lo que hace necesaria la segunda solución.

Bajo el supuesto de que el coste computacional está directamente relacionado con el tamaño de la imagen, esta solución consiste en un cambio del formato utilizado por uno de

resolución menor comprobando que el tiempo entre adquisiciones disminuye y también que este cambio no perjudica la precisión de adquisición de datos de FaceAPI.

Primero, se ha comprobado que realmente el cambio efectúa un aumento de la frecuencia de muestreo tal y como lo demuestra la siguiente gráfica (fig. 5.4). Es una representación del histograma del tiempo transcurrido entre frames. A medida que la resolución baja, el tiempo entre adquisiciones también decrece, el máximo pasa a ser de 35ms en la más alta resolución a casi de 30ms para la resolución menor, además la concentración de tiempos en torno al máximo es mayor para una resolución menor.

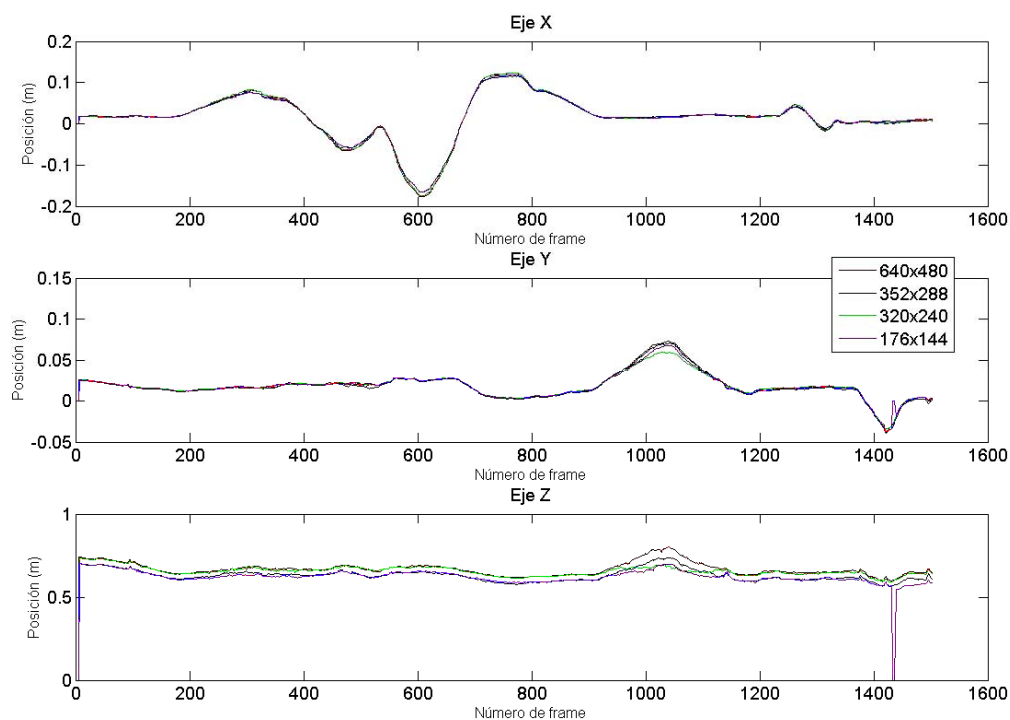
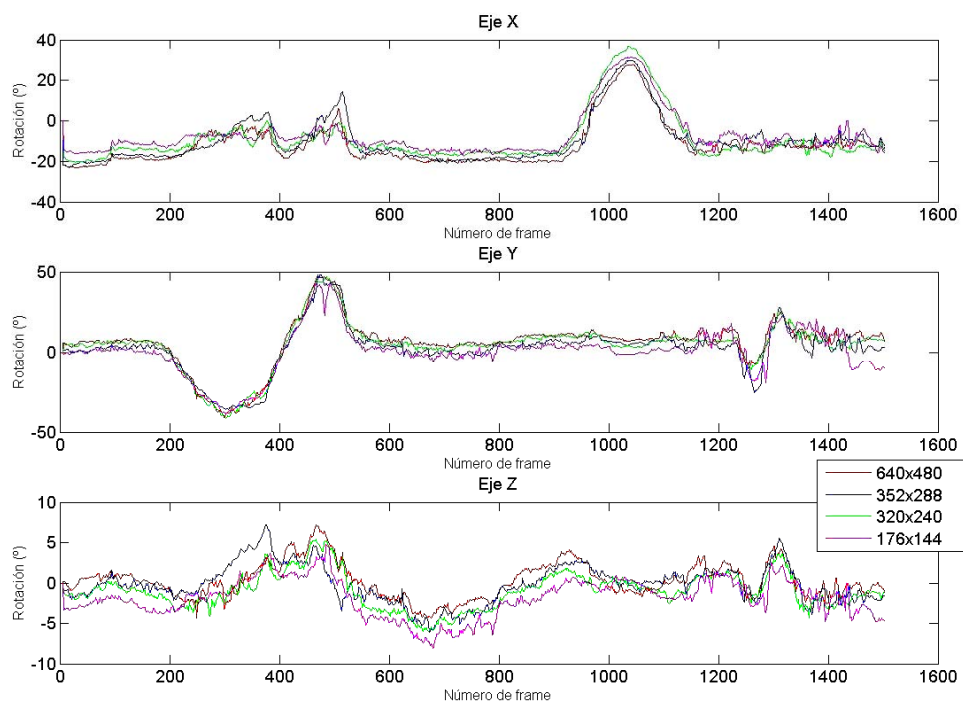


**Fig 5.4** Histograma de la diferencia entre los tiempos de adquisición consecutivos.

Una vez validada esta opción como solución se va a comprobar si el cambio va a dañar la precisión de FaceAPI. Esta prueba ha sido elaborada mediante el uso de la aplicación de FaceAPI que calcula la posición y rotación de la cabeza de un usuario en vídeos. Esta aplicación nos permite estudiar un mismo vídeo en el que se contemplan diferentes desplazamientos y rotaciones posibles con alguna combinación de los mismos, de diferentes resoluciones.

Concretamente vamos a grabar un vídeo en resolución estándar (la resolución por defecto, es decir, 640x480 píxeles) y mediante un programa de cambio de formatos convertirlo a vídeo de resolución menor: a uno de 352 x 288 píxeles, a otro de 320 x 240 píxeles y a otro de 176 x 144 píxeles. Naturalmente, todos los formatos son admitidos por FaceAPI.

En las siguientes gráficas (fig. 5.5, 5.6, 5.7 y 5.8) se muestran los resultados obtenidos de un primer vídeo. Las cuatro primeras figuras muestran la evolución de la posición de la cabeza, la rotación de la cabeza, la posición del ojo derecho y la posición del ojo izquierdo consecutivamente a medida que el vídeo avanza. Cada una de estas figuras está compuesta de tres gráficos ya que cada una de las variables está compuesta de tres componentes correspondientes a las coordenadas X,Y,Z. En cada una de las gráficas están diferenciadas en 4 colores distintos las resoluciones del vídeo utilizado.

**Fig. 5.5** Posición de la cabeza**Fig. 5.6** Rotación de la cabeza

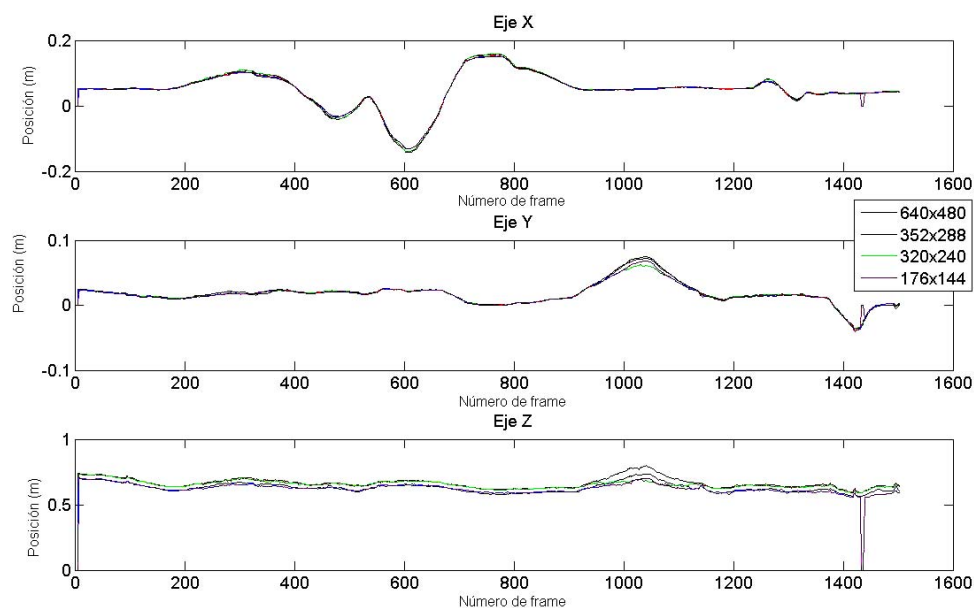


Fig. 5.7 Posición ojo derecho

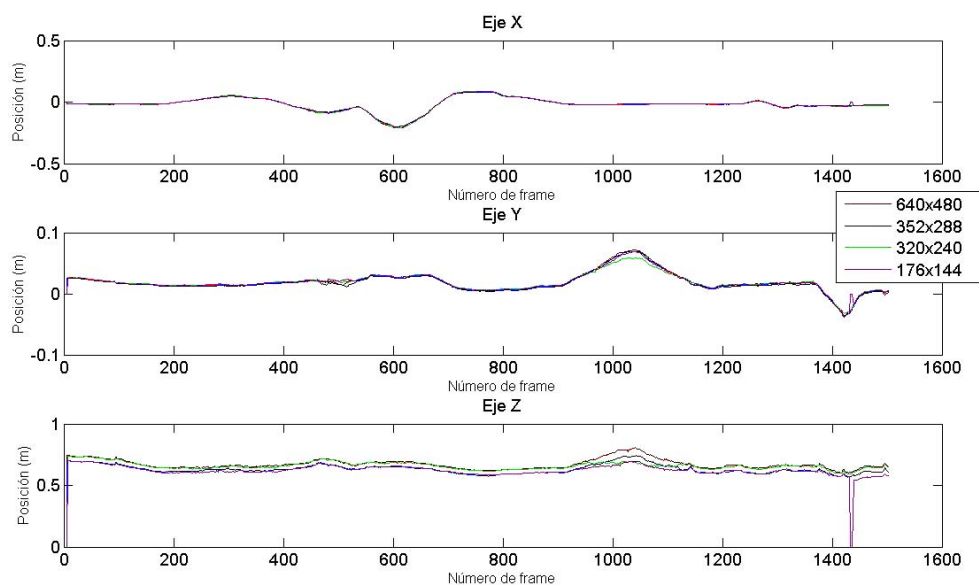


Fig. 5.8 Posición ojo izquierdo

En las gráficas que muestran la posición del usuario del vídeo, ya sea de la cabeza o de los ojos, se observa un comportamiento similar entre las tres figuras (ver fig 5.5, 5.7 y 5.8). En los tres casos, en los gráficos de la posición en los ejes  $X$  e  $Y$ , la resolución del vídeo no implica diferencias notables exceptuando en los desplazamientos extremos en los que se nota una pequeña diferencia. Sin embargo, las diferencias en el eje  $Z$  son mayores para todo el vídeo. Es curioso que ante un movimiento extremo en el eje  $Y$  el cambio de resolución afecta también al

cálculo de la posición en el eje Z. Las tablas V y VI muestran la diferencia existente entre los vídeos de cada resolución estudiada con el que trabaja FaceAPI por defecto, es decir, el de 640 x 480 píxeles de resolución para poder cuantificar numéricamente si este cambio de formato afectaría la precisión de FaceAPI.

Diferencia máxima entre las resoluciones estudiadas con la de 640x480			
Formatos de los vídeos	Diferencias de posición (cm.)		
	EJE X	EJE Y	EJE Z
352 x 288	1.22	0.5	6.83
320 x 240	1.70	2.40	1.10
176 x 144	1.70	2.86	4.7

**Tabla V.** Diferencia máxima de posición entre los vídeos de diferentes resoluciones y el de 640x480.

A partir de los datos anteriores (tabla V) observamos que el comportamiento de los vídeos en más bajas resoluciones a la utilizada por defecto es aceptable, es decir, no hay diferencias significativas en el cálculo de la posición de la cabeza exceptuando el eje Z, el que presenta más problemas. Considerando que el error cometido por FaceAPI es de 1cm las diferencias máximas obtenidas en los ejes X e Y no son muy altas ya que la diferencia es dos veces este error (aunque en el vídeo de más baja resolución es más alta) pero la diferencia media en el eje X no llega a los 0,3cm, en el eje Y a los 0,2cm y en el eje Z a los 0,5cm. (Esta información de la diferencia media no la ofrecemos en la tabla).

En la figura correspondiente a la rotación de la cabeza (ver fig. 5.6), la rotación calculada por cada uno de los cuatro vídeos sigue la misma línea de tendencia, sin embargo, teniendo en cuenta que el error cometido por FaceAPI al calcular la rotación el error es de 3° el cambio de formato afectaría más.

Diferencia máxima entre las resoluciones estudiadas con la de 640x480			
Formatos de los vídeos	Diferencias de rotación (°)		
	EJE X	EJE Y	EJE Z
352 x 288	3.2	4	6.35
320 x 240	9	5.5	11,2
176 x 144	14	11	6,27

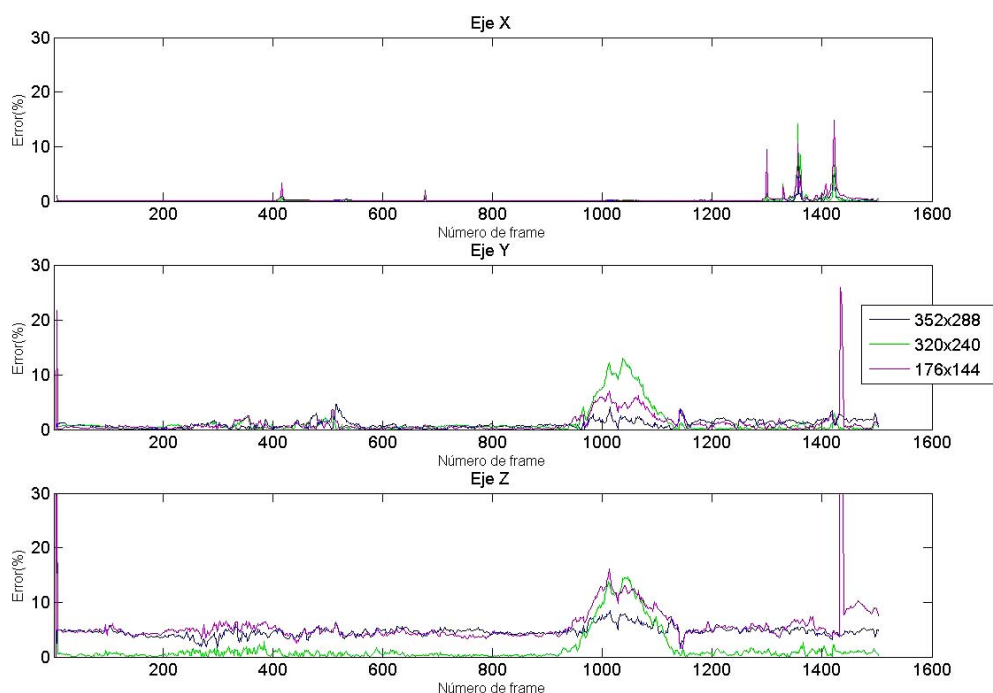
**Tabla VI.** Diferencia máxima de rotación entre los vídeos de diferentes resoluciones y el de 640x480.

Las diferencias mostradas en la tabla son las diferencias máximas del comportamiento general ya que en este caso se forman ciertos picos correspondientes a un solo frame de más altas diferencias. En general las diferencias aumentan al producirse cambios de posición de la cabeza críticos, próximos a los máximos desplazamientos que puede medir FaceAPI.

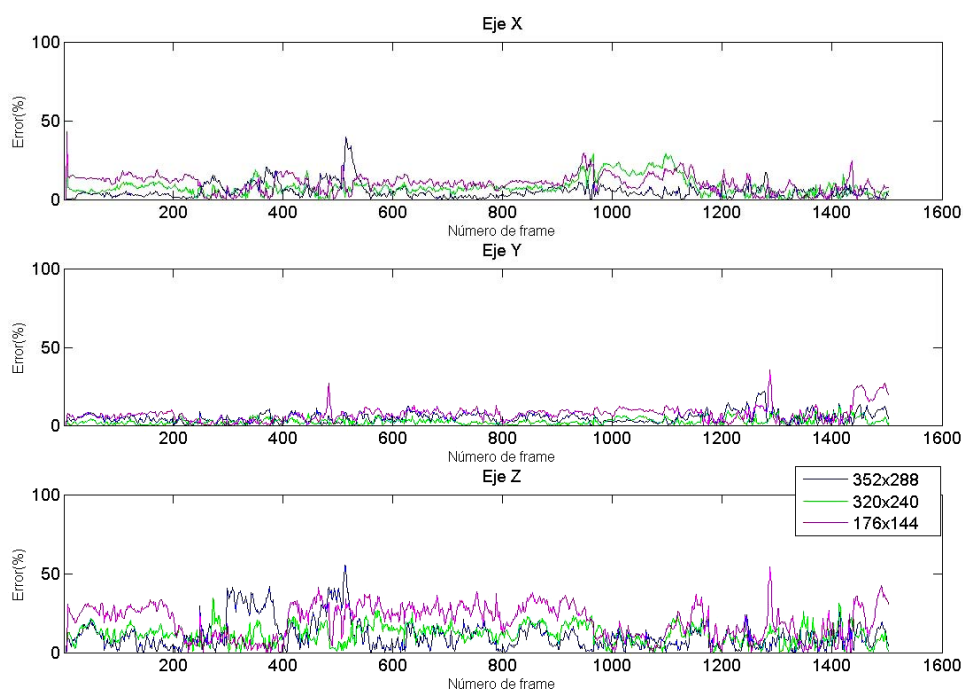
Se omiten las tablas que presentan las diferencias máximas de posición de los ojos de los vídeos de diferentes formatos con el original ya que se comportan de manera muy parecida a los datos de la posición de la cabeza.

Para una mejor apreciación de la diferencia de los datos de cada uno de los vídeos se ha calculado el error cometido por los vídeos de diferente resolución respecto al original. El error ha sido calculado como la resta entre la información (tanto para la posición como para la rotación de la cabeza) del vídeo original menos la del vídeo del que queremos calcular,

normalizado entre la variabilidad de los datos del vídeo original multiplicado por 100 para obtenerlo como un porcentaje. En este caso solo se muestran las gráficas correspondientes al error en la posición y rotación de la cabeza (ver fig. 5.9 y 5.10) ya que las correspondientes a la posición de los ojos ofrecen un comportamiento similar al del error de posición de la cabeza.



**Fig. 5.9** Error relativo a la posición de la cabeza.



**Fig. 5.10** Error relativo a la rotación de la cabeza.



El error observado en la figura 5.9 es prácticamente despreciable en los ejes  $X$  e  $Y$  salvo tres picos en la primera gráfica probablemente debidos a algún movimiento crítico (por ejemplo un movimiento en el límite de los controlados por FaceAPI) y en el segundo gráfico un pico en el vídeo de más baja resolución probablemente un fallo del vídeo ya que coincide con un pico de la primera y tercera gráfica. Sin embargo, en el gráfico con información correspondiente al eje  $Z$  (fig 5.9 tercer gráfico) el error aumenta para dos casos: el vídeo de 352x288 y el de 176x144 píxeles de resolución, aunque el error del vídeo de 320x240 píxeles de resolución sigue manteniéndose bajo a pesar de tener menor resolución que el primero de los anteriores. De los frames 950 a 1100 de los vídeos vemos que el error aumenta considerablemente al mismo tiempo que lo hace en el eje  $Y$ , esto también pudiera ser debido a un desplazamiento mayor al máximo que puede calcular FaceAPI.

La variación de la rotación respecto al error de rotación asumido por FaceAPI en general es superior a la variación de la posición respecto al error de posición especificado por este sistema por ello principalmente nos vamos a centrar en la información de la posición de la cabeza para la sincronización con FaceAPI. Este comportamiento al igual que en el caso del error de la posición se ve incrementado en la rotación relativa al eje  $Z$ .

A la vista de los resultados anteriores vamos a cambiar de formato para que FaceAPI funcione a tiempo real. A pesar de que existen ciertas diferencias entre el cálculo de la posición y orientación de la cabeza del mismo vídeo con diferentes resoluciones, en general estas no superan 1cm, el error cometido por FaceAPI. Las diferencias entre estos vídeos aumentan ante movimientos abruptos los cuales no serán frecuentes al utilizar Shakti. Por tanto utilizaremos un formato de resolución de 320 x 240 píxeles, la resolución intermedia entre las tres probadas ya que consigue mantener un frame rate cerca de 30 muestras por segundo más fácilmente que el formato con resolución mayor (352 x 288) y el error respecto al formato original no es tan alto como el de resolución menor (176 x 144).

#### 5.4.2 Desplazamiento en el eje $Z$ .

Como ya se ha comentado anteriormente (ver apartado 5.2), existe un problema al medir los desplazamientos de usuarios distintos ya que FaceAPI no considera que diferentes usuarios estén en la misma posición aunque realmente lo estén.

Para llegar a esta conclusión se optó por realizar una prueba utilizando la ayuda de fotografías de usuarios en lugar de usuarios reales y jugar así con la posibilidad de aumentar o disminuir el tamaño de dichas imágenes. La prueba es muy simple: se trata de utilizar una fotografía de una persona en tamaño pequeño sobre un soporte a una distancia fija para almacenar los datos de la posición de la misma y más tarde, aumentar el tamaño de la misma fotografía y colocarla en la misma posición a la anterior. El resultado es inmediato: a pesar de que la posición en el eje  $X$  y eje  $Y$  coincidan en ambos casos, el resultado en el eje  $Z$  es claramente diferente.

Del resultado anterior se concluye que FaceAPI trabaja con algún modelo matemático interno e intuye la posición del eje  $Z$  (eje cámara nariz) a partir del tamaño de la cara o algún rasgo facial ya que FaceAPI solamente tiene información de dos dimensiones por utilizar una única webcam.

Entonces, partiendo de que la posición relativa al eje  $Z$  sería diferente para cada usuario, se intentó comprobar si la diferencia entre dos posiciones fijas de dos usuarios distintos son idénticas y por tanto aunque la posición absoluta no, los desplazamientos serán equivalentes.

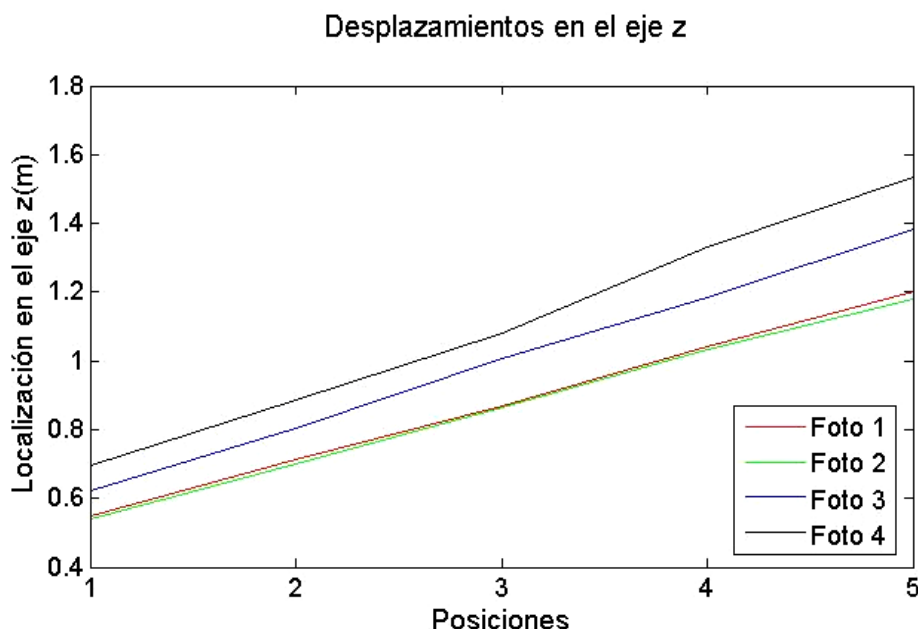
Para ello nuevamente realizaremos una primera prueba con fotografías de usuarios, y dado que el eje  $Z$  es el más crítico ya que es el único eje que con usuarios en la misma posición



la localización deducida por FaceAPI es distinta, solo se considerarán desplazamientos en este eje.

La prueba consiste en que desde un punto de partida fijo para todas las fotografías se avanzarán 10cm hacia delante, más tarde otros 10cm, después 30cm hacia atrás y por último otros 10cm atrás. Por lo que tendremos información de la posición de la cabeza en 5 posiciones a lo largo del eje Z, concretamente en un rango de 50cm. Puesto que FaceAPI no puede realizar el seguimiento de más de una cara al mismo tiempo las pruebas se harán usuario por usuario.

En el gráfico de la figura 5.11 se muestra la posición de la cabeza en cada una de las cinco posiciones ordenadas de la más cercana a la más alejada a la cámara. Claramente no se considera que todos los usuarios pasen por los mismos puntos aunque todos ellos siguen un comportamiento casi lineal con la distancia. Sin embargo, los usuarios 2 y 4 si coinciden en la posición del eje Z en todas las posiciones ya que la diferencia entre ambas nunca supera los 2cm y el error que puede cometer FaceAPI al determinar la posición de un usuario es de 1cm.



**Fig. 5.11** Desplazamientos controlados en el eje Z de usuarios reales

Este comportamiento deja de parecer lineal en las últimas posiciones ya que el error de FaceAPI a distancias más alejadas aumenta. Pese a ello, se podría decir que todos los desplazamientos en el eje Z son de 0.12 unidades de FaceAPI contando con el error acometido por el software. Además, el propio sistema de realización de las pruebas también cuenta con un error propio debido al cambio de posición del soporte o chinrest en el que se realizan las mismas. Este error ha sido cuantificado recolocando en una misma posición el soporte con una misma fotografía múltiples veces y calculando la variabilidad de la posición de la misma; nuevamente es de alrededor a un 1cm.

En este punto, se intenta realizar otras pruebas semejantes a las anteriores para usuarios reales. Pero, antes de eso se pretende averiguar si existe alguna relación entre alguno de los parámetros antropométricos y la posición calculada en el eje Z.

El primer parámetro que se estudió fue la distancia entre los ojos obtenida por la diferencia entre las componentes  $X$  de las variables de posición de los ojos para observar si a partir de esta información se deduce la posición en el eje  $Z$ . La comprobación consiste en estudiar el comportamiento de la distancia entre los ojos realizando un desplazamiento de un usuario constante por el eje  $Z$ . El resultado de esta prueba es desfavorable ya que la diferencia entre las componentes  $X$  de las variables de posición de cada ojo, es decir, la distancia entre los ojos es constante independientemente de la posición en  $Z$ . Esto podría ser debido a que la posición de los ojos determinada por FaceAPI es calculada a partir de la combinación de la información de la posición y la rotación de la cabeza, normalizado respecto a un modelo matemático interno.

Puesto que la información anterior de los ojos no parece que esté directamente relacionada con la posición del usuario en el eje  $Z$ , pues es constante todo el tiempo, debemos recurrir a fuentes de información alternativas disponibles en FaceAPI: la variable que incluye información de los puntos de los contornos de los rasgos de la cara. Esta variable como ya se ha mencionado en la introducción a FaceAPI contiene información de 4 esquinas de los ojos y de los centros de la pupila entre otras, por lo que esta información también podría suponer el punto de partida para calcular la posición en el eje  $Z$ .

En la imagen 5.12 se observan los 8 puntos que calcula FaceAPI del contorno de los ojos además de los centros de la pupila. De la misma manera que hicimos al estudiar la distancia entre la posición de los ojos en el eje  $X$ , estudiaremos la separación entre los ojos dada por la posición de los rasgos de la cara frente a desplazamientos en  $Z$  de diferentes usuarios. Concretamente calcularemos la posición central de cada ojo a partir de la media entre las posiciones de los dos extremos de cada ojo, y más tarde, calcularemos la distancia entre los ojos a partir de los cálculos anteriores.



**Fig. 5.12** Landmarks

Para ello, además del archivo de texto que guardamos con la información relativa a la posición y rotación de la cabeza, crearemos ahora un nuevo archivo donde almacenaremos algunas de las componentes de la variable que contiene información de la posición de los rasgos más destacados de la cara: 5 puntos de cada ojo y la información relativa al tiempo para poder sincronizarlo con la información de la posición en el eje  $Z$ . De esta manera podremos poner en común estas variables y observar si existe una correlación entre ellas.

Una vez realizadas las modificaciones necesarias en el código de FaceAPI empezaremos con las pruebas mencionadas con usuarios reales. En estas pruebas los usuarios han sido colocados en cuatro posiciones diferentes distanciadas 10cm cada una de ellas a lo largo del eje  $Z$ . Estas posiciones serán controladas mediante el chinrest manteniendo al usuario con la cabeza quieta para una mayor precisión.

A partir de estas pruebas se extraen los datos relativos de la posición de la cabeza en el eje  $Z$  y los landmarks de los ojos en cada una de las posiciones donde ha sido posicionado el usuario. De cada posición se ha promediado tanto la información de la posición en el eje  $Z$  como la de la distancia entre los reflejos y se ha calculado la desviación de cada una de las variables. En los siguientes gráficos (fig 5.13 y 5.14) se muestra la distancia de los ojos media en cada una de las posiciones (línea continua) delimitada por la desviación correspondiente (líneas discontinuas). El eje  $X$  no representa nada simplemente se utiliza para una mejor visualización del intervalo.

El objetivo último es caracterizar y conocer bien cómo FaceAPI modela el desplazamiento en Z. En este sentido, se trata de comprobar si existe una relación entre este desplazamiento y la distancia entre ojos, ya que pudiera ser un parámetro interesante de diseño a incorporar en la ecuación de calibración. Después de analizar la posición de ojos que FaceAPI da por defecto, se ve que estos están normalizados y por tanto no proporcionan información relevante, por lo que se recurre a esta otra información de la variable que aporta la posición de las features de la cara (posición de los landmarks).

Para todos los usuarios parece que en cada una de las posiciones en Z donde se ha calculado la separación entre los ojos, FaceAPI encuentra una separación distinta de los mismos ya que los intervalos de la distancia entre los reflejos de cada posición no se solapan exceptuando dos casos. Sin embargo, estas medidas no son muy fiables ya que al realizar el seguimiento en posiciones consecutivas en todos los casos se ha perdido el seguimiento y podríamos contar con un error de 1cm. Además, no se percibe una disminución de la distancia conforme avanzamos de posición desde la primera a la cuarta, ya que las posiciones están ordenadas desde la posición más cercana a la cámara con desplazamientos consecutivos de 10cm hacia atrás. Es decir, con estas pruebas (ver fig. 5.13) se comprueba que existe una variación de la distancia entre ojos en cada posición, pero no está clara su relación directa con la posición en el eje Z.

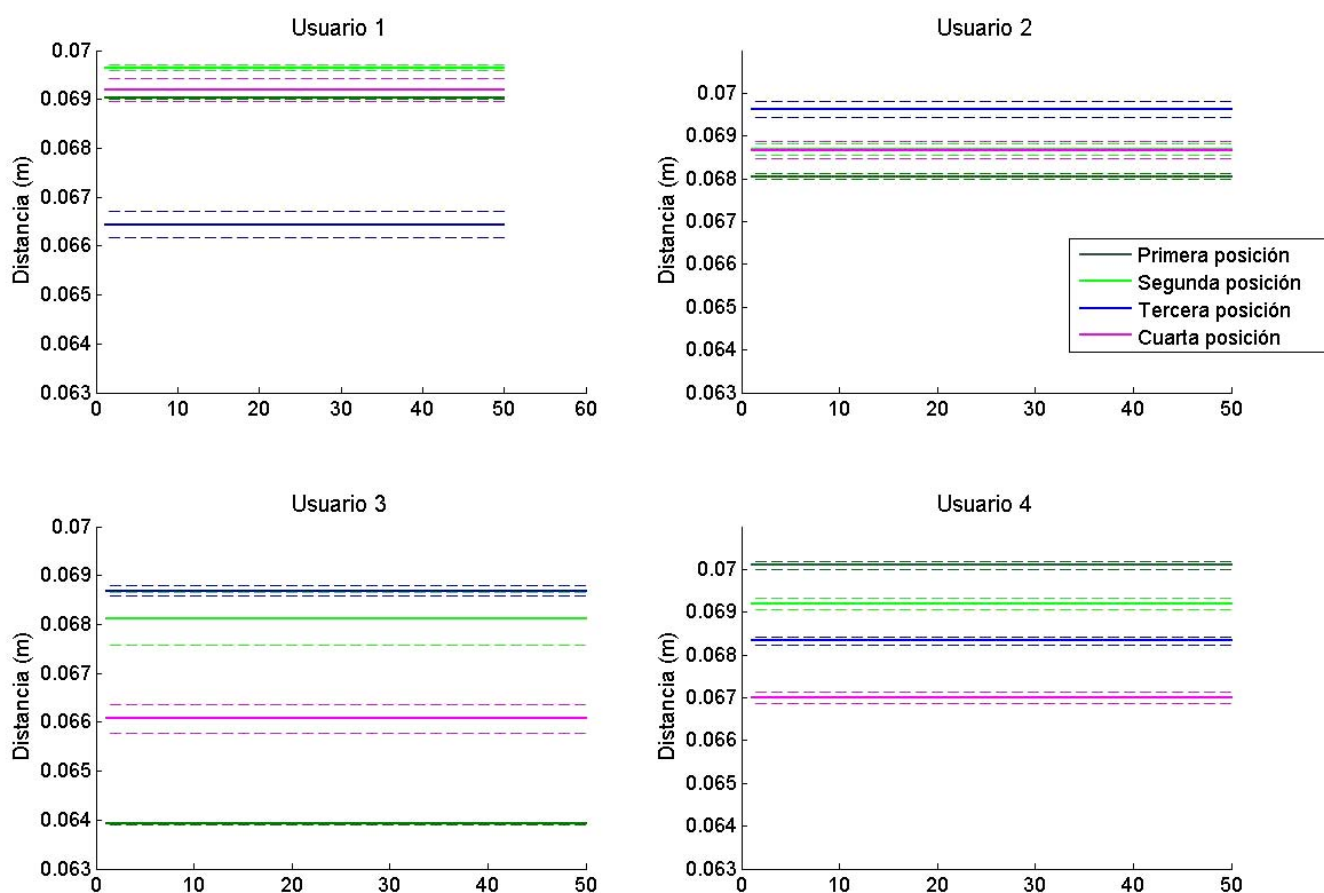


Fig. 5.13 Distancia entre los ojos en cuatro posiciones diferentes en Z

Además de la información de la separación de los ojos estas pruebas nos permiten comprobar si los desplazamientos en el eje  $Z$  para distintos usuarios son medidos de la misma manera.

Estos datos (ver fig. 5.14) muestran diferencias en el comportamiento de FaceAPI entre usuarios. Sin embargo, se observan similitudes entre los usuarios 1 y 2 y los usuarios 3 y 4. Los dos primeros se solapan en primera y tercera posición lo cual indica de que FaceAPI considera que están en el mismo punto y en los otros dos usuarios también se solapan pero en las dos primeras posiciones. La última posición es más problemática teniendo en cuenta que el error que cometerá FaceAPI es mayor.

No obstante con estas pruebas observamos que la información que nos ofrece FaceAPI de la posición de la cabeza en el eje  $Z$  es poco fiable ya que es muy imprecisa. Además después de realizar un intento de observar la separación entre los ojos como un indicativo de la posición en el eje  $Z$  vemos que esta distancia es muy parecida en cualquier posición y contando con que el error de FaceAPI es de 1cm esta información no sería muy útil.

Aunque FaceAPI no proporcione información fiable sobre la localización real de la cabeza del sujeto, es posible que tanto esta variable (posición de cabeza) como la separación entre ojos ofrezcan información potencialmente útil para la optimización de la ecuación de calibración. Es importante considerar que la calibración de un sistema de eye-tracking como el nuestro es personal e independiente para cada usuario (ver apartado 2.1). Con esto, poco importa que la información sea objetivamente fiel o se corresponda con la realidad ya que lo realmente importante es el comportamiento intra-usuario. Esto es, puede que la separación entre ojos de FaceAPI para un usuario resulte todavía de utilidad para corregir la calibración específica de dicho sujeto. Se analizará más adelante.

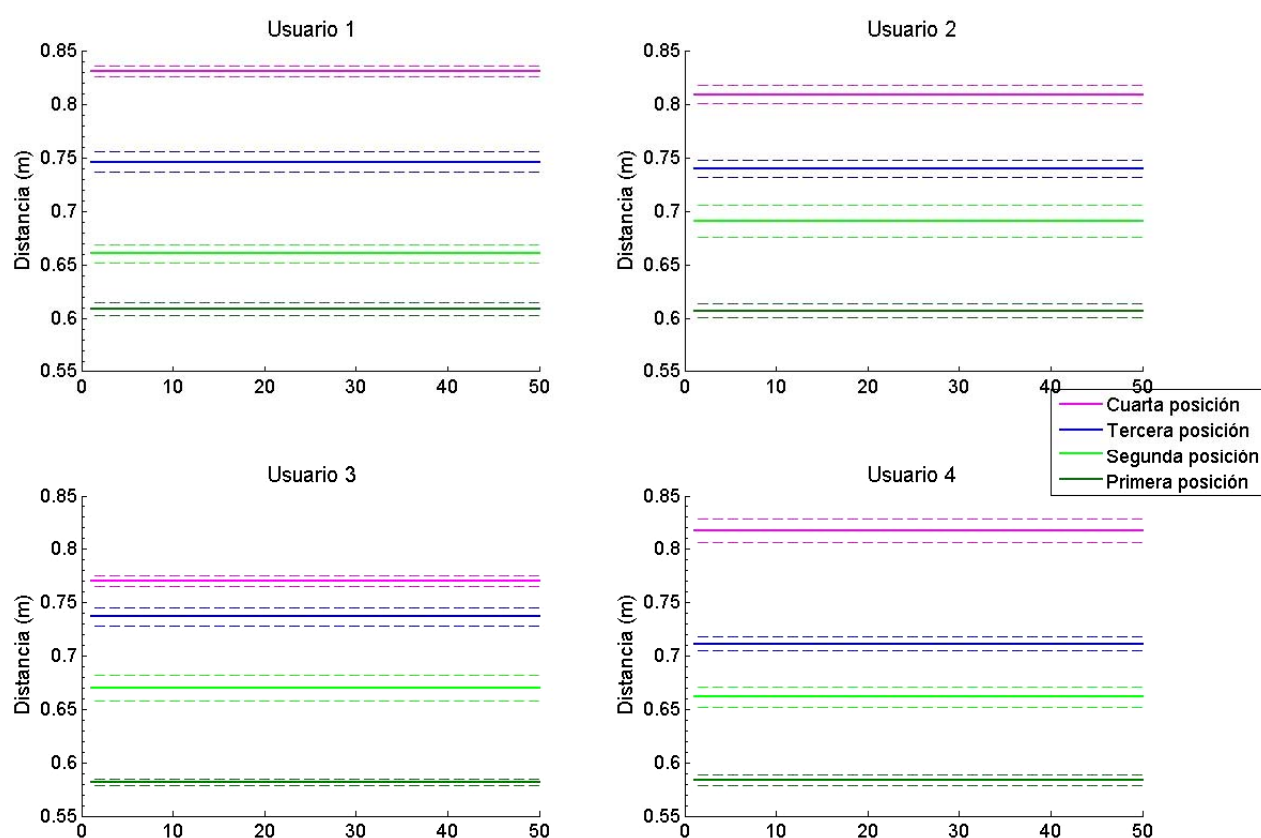


Fig. 5.14 Posición en el eje  $Z$

## 5.5 Conclusiones

FaceAPI es una herramienta que calcula la posición y orientación espacial de la cabeza (o del ojo) por lo que potencialmente esta información puede ser complementaria al sistema de eye-tracking para aumentar la robustez de este sistema frente a movimientos del usuario.

Para poder integrar la información de FaceAPI en Shakti se quiere realizar una sincronización de ambos sistemas y para ello extraeremos dicha información relativa a la posición y orientación de la cabeza junto con información del tiempo de cada captura.

La información del tiempo añadida corresponde con el tiempo transcurrido desde el último encendido del equipo hasta el instante de la captura correspondiente.

A partir de la información del tiempo se realizó un estudio de las muestras por segundo que obtenía FaceAPI y se observó que era baja no permitiendo el funcionamiento a tiempo real (30 frames segundo). Por ello, se realizó un cambio de formato para utilizar una resolución de las imágenes de vídeo de 320 x 240 píxeles consiguiendo así el funcionamiento a tiempo real, requisito indispensable en Shakti.

Por otro lado, las unidades de la información relativa a la posición de la cabeza son la distancia respecto a la cámara en metros y la orientación en grados. En este caso, realizando un estudio del comportamiento de FaceAPI al calcular la posición de la cabeza hemos encontrado cierta imprecisión al calcular la posición del usuario en el eje Z. Realizando el tracking de varios usuarios en la misma posición FaceAPI no considera que todos se encuentren en la misma posición ofreciendo diferencias entre usuarios muy superiores a 1cm, que según las especificaciones es el error máximo que comete esta herramienta. A partir de este resultado deducimos que FaceAPI puede que trabaje con un modelo matemático interno ya que para obtener la localización espacial de un objeto a partir de un sistema sin calibrar de una única cámara, es necesario recurrir a un modelo matemático-estadístico previo. Es posible que a partir de una serie de parámetros de la imagen (medidas antropométricas como la distancia entre los ojos, la nariz, etc.) se realice algún tipo de calibración interna del modelo. De este modo, la distancia entre el usuario y la cámara corresponderá a aproximaciones realizadas a partir de dicho modelo.

A partir de estos resultados se empieza a poner en duda la herramienta FaceAPI por su imprecisión al calcular la posición en el eje Z. Sin embargo, se va a intentar relacionar esta información con Shakti y realmente en ese punto se cuestionará la validez del sistema para el objetivo perseguido.

## 6 Estudio experimental

Como hemos visto anteriormente la ecuación de calibración utilizada en el software de seguimiento de la mirada presenta una especial sensibilidad ante desplazamientos del usuario [1]. El objetivo del presente proyecto será tratar de corregir la ecuación de partida con el objetivo de minimizar el error proporcionando una mayor libertad de movimientos al usuario. Para ello es necesario, en primera instancia, caracterizar el error producido por esta ecuación frente a desplazamientos controlados, y a partir de ahí, viendo la tendencia del error de cada movimiento, intentar corregir la ecuación de calibración.

Siguiendo la estrategia presentada por Cerrolaza et al. [1], la optimización de una ecuación de calibración ante los desplazamientos del usuario pasaría por el estudio de todas las posibles expresiones matemáticas construidas a partir de las variables predictivas del sistema. La evaluación de estas ecuaciones en las diferentes posiciones nos permitiría hallar aquella que minimizara el error ante los desplazamientos. La filosofía de trabajo que se adoptará en este proyecto será la de tratar de caracterizar una ecuación de calibración inicial (apartado 3), estudiando su comportamiento ante los desplazamientos y proponiendo métodos de optimización y corrección más heurísticos y directos que los basados en la ‘fuerza bruta’. Obviamente si la primera opción de mejora no demuestra tendencias claras o un resultado aceptable, se debería de considerar esta alternativa.

En este estudio participarán un total de doce usuarios para medir el error cometido por la ecuación de partida ante desplazamientos de los usuarios. En comparación con estudios anteriores [3], [4], [6], [9], el número de usuarios es muy elevado, dotando al estudio de gran relevancia práctica.

La prueba consiste en calibrar en una serie de posiciones controladas, las cuales más tarde serán especificadas tomando como referente una posición central. El objetivo será estimar el  $PoR$  en cada una de las posiciones en las que se ha realizado una calibración utilizando para ello los coeficientes calculados a partir de la información de la calibración en la posición central y obtener así el error debido al desplazamiento.

### 6.1 Posibles movimientos a estudiar.

La primera decisión que tenemos que tomar para plantear este estudio es los desplazamientos de la cabeza que vamos a considerar. Siguiendo el objetivo del proyecto, es decir, prever si existe relación entre el error cometido por la ecuación de calibración de partida y un desplazamiento en concreto del usuario, optamos por estudiar desplazamientos en el eje Z. Se toma esta decisión ya que como estudios anteriores advierten [9] y [19], los movimientos en el eje Z son los más críticos, es decir, los que mayor error presentan.

En un principio, se intentaron abarcar todos los movimientos de la cabeza posibles y todas sus combinaciones. Teniendo en cuenta que los movimientos más significativos que FaceAPI puede calcular se obtienen mediante la posición y la rotación de la cabeza, los posibles movimientos a estudiar eran 6 pero la combinación de estos ascendería a un número muy alto de posiciones donde realizar una calibración. Asumiendo la imposibilidad de realizar estas pruebas por ser excesivamente largas y muy difíciles de conseguir con el sistema utilizado en el proyecto, se realizó un filtrado del número de movimientos que se querían estudiar.

Entonces, se intentó averiguar si realmente eran necesarias las medidas de rotación además de las de posición ya que bajo la suposición de que el ojo es esférico, una rotación equivaldría o podría interpretarse como la combinación de dos traslaciones.

### 6.1.1 Filtrado de movimientos posibles a estudiar

En ese momento se intentó demostrar si es cierto que una rotación equivale a una traslación, porque eso permitiría limitar el estudio de las posiciones de cabeza a estudiar. Por ello se decidió realizar una aplicación con la ayuda de FaceAPI para monitorizar dos calibraciones de un mismo usuario, ambas manteniendo el ojo en la misma posición. Sin embargo en la primera posición deberá existir rotación de la cabeza y en la segunda no.

Manteniendo la cabeza en una posición concreta (primer paso), FaceAPI calculará la rotación de la cabeza media de un periodo de tiempo controlable y almacenará dicha información para una posterior utilización.

Después, con una rotación de la cabeza del usuario (segunda paso), nuevamente FaceAPI almacenará la información de la posición media de la cabeza de otro breve periodo de tiempo. Entonces, manteniéndose el usuario en la misma posición se realiza una calibración de 16 puntos.

Por último, el usuario deberá intentar ponerse en la posición guardada en el segundo paso y la rotación de la cabeza deberá ser la misma que en el primero. Para este ajuste se considera un error de 2 veces la desviación estándar para cada componente de los datos obtenidos por FaceAPI al intentar mantener la cabeza quieta sin la utilización del chinrest. Esta prueba no requiere del uso del reposa cabezas ya que podría dificultarla. Una vez que el usuario consiga llegar a esta posición con la rotación de la cabeza requerida la aplicación lo indicará por pantalla. Manteniendo también la misma posición de la cabeza, se va volverá a calibrar y se dará la prueba por finalizada.

Esta prueba permite controlar que la rotación realizada en el segundo paso y la traslación realizada en el tercero coincidan en la posición espacial. Y los datos obtenidos en las respectivas calibraciones nos dan el resultado de que ambas opciones nos llevan al mismo resultado. Es decir, la diferencia máxima entre los datos obtenidos en ambas calibraciones se diferencian en unos 50 píxeles, lo cual equivale a una distancia de las features del ojo de 0,37mm.

### 6.1.2 Alcance del estudio

En una segunda instancia, a partir del filtrado de los movimientos de la cabeza a estudiar solo nos quedaban los movimientos de traslación y posibles combinaciones de los mismos. Entonces, pensamos en hacer las pruebas en 27 diferentes posiciones (es decir, un cubo 3x3 donde se estudiarían todas las combinaciones de las traslaciones en los tres ejes).

Sin embargo, este estudio aunque abarcaría todos los movimientos de la cabeza no da una idea muy clara de la tendencia del error que comete Shakti al avanzar en un solo eje. Por tanto, se plantea la idea de realizar pruebas en cada eje por separado a partir de una posición central avanzando hacia delante y hacia atrás, hacia arriba y hacia abajo o en su caso hacia un lado o al otro tomando más de tres puntos en cada uno de los ejes para poder ver una tendencia en el error si esta existe.



Ambas opciones de análisis del error producido por todos los movimientos considerados suponen una larga preparación de las pruebas y también mucho tiempo de realización con cada uno de los usuarios. Por tanto, en este proyecto hemos decidido reducir este estudio y únicamente analizar el comportamiento de la ecuación que partimos con traslaciones en el eje  $Z$  ya que se considera el más crítico [19] y se ha preferido realizar un estudio más reducido con más usuarios y así asegurar la validez del resultado. Sin embargo, el número de puntos en donde se realizarán las diferentes calibraciones serán decididos a la hora de la preparación de las pruebas ya que están limitadas por el rango de movimientos en los que Shakti pueda mantener el seguimiento. Es decir, el rango de desplazamiento en el plano imagen, una iluminación óptima en todas las posiciones, etc.

## 6.2 Sistema utilizado para las pruebas

El sistema utilizado para las pruebas se diferencia en algunas cosas del sistema descrito anteriormente tal y como se ha venido estudiando hasta ahora. Estas diferencias han sido necesarias para permitir el movimiento del usuario en dos aspectos.

La primera dificultad la encontramos en el objetivo de la cámara utilizada porque al ser de 35mm los desplazamientos del usuario quedan invalidados ya que el ojo con pequeños movimientos saldría del plano imagen y la cámara tiene que permanecer en la misma posición y orientación en las pruebas. En vista a esto, decidimos utilizar el objetivo de 16mm que permitirá mayor rango en los desplazamientos.

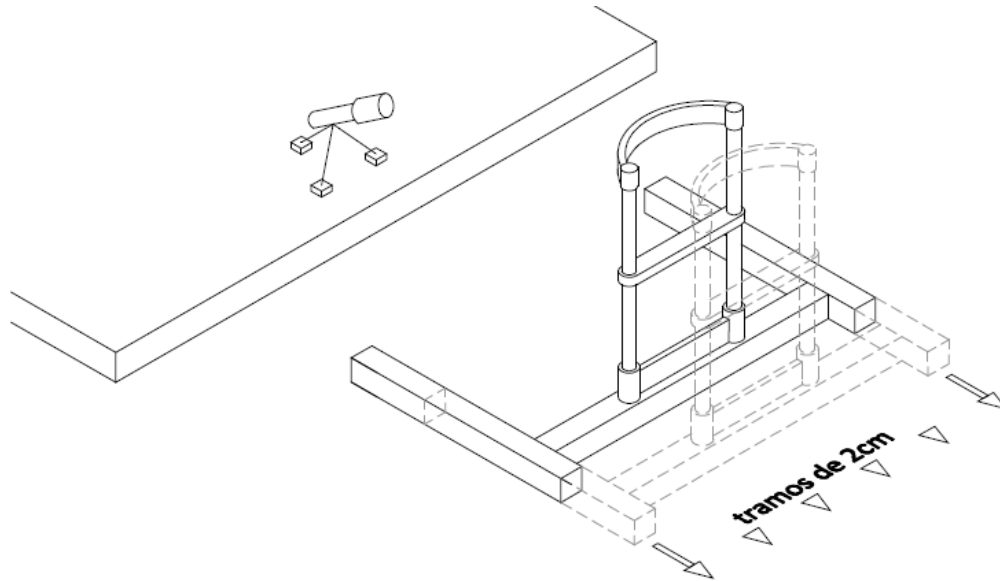
Una vez elegido el objetivo de la cámara realizamos pruebas con ayuda del chinrest estudiando para diferentes posiciones relativas entre la cámara y el usuario, y orientación de la cámara el recorrido máximo realizable por el usuario sin que Shakti pierda el ojo del plano imagen. Pero, este no es el único problema ya que a pesar de que el ojo se encuentre en el plano de imagen, Shakti debe ser capaz de realizar el seguimiento del ojo en una trayectoria podamos realizar calibraciones en más de cinco puntos.

Al intentar comprobar el funcionamiento de Shakti nos percatamos de que el seguimiento no era posible debido a que en algunas posiciones el tamaño de la pupila o de los reflejos era insuficiente y sobre todo porque en el código existe una limitación en la distancia entre los reflejos mínima que en este caso es menor a la preestablecida. Todo ello es causa del cambio del objetivo, se reduce el tamaño de la pupila, los reflejos y la distancia entre estos. El tamaño de la pupila y reflejos mínimo no presenta mucha complicación ya que pueden modificarse en los parámetros de seguimiento de cada usuario de manera sencilla, sin embargo, la distancia entre los reflejos mínima no. Por tanto, se decide separar los iluminadores de la pantalla provocando así un aumento de esta distancia. La posición de los iluminadores pasa a constituir un grado de libertad más del sistema, permitiendo un ajuste diferente por cada usuario.

### 6.2.1 Alcance del recorrido por el eje $Z$

A partir de estos cambios se consigue mantener el seguimiento del ojo por Shakti en una distancia de 12cm realizando modificaciones en los parámetros de seguimiento de los usuarios y colocando los iluminadores estratégicamente. Por tanto, se decide realizar pruebas en siete posiciones diferentes espaciadas por 2cm cada una a lo largo del eje  $Z$  siguiendo el esquema representado en la figura 6.1.





**Fig. 6.1** Esquema del desplazamiento entre posiciones.

#### 6.2.1.1 Posición de los iluminadores

En este apartado se van a definir algunas pautas que se han seguido para la correcta colocación de los iluminadores y así reducir la precisión del ajuste necesario de los parámetros de seguimiento para un correcto funcionamiento en todas las posiciones. Ya que estos parámetros deberán mantenerse constantes para cada usuario así como la posición de los iluminadores.

Para colocar correctamente los LEDs habrá que enfocarlos directamente a la cara del usuario teniendo en cuenta que esta orientación deberá iluminar al usuario en todas las posiciones. Pero, será más importante que en la posición más alejada estén lo mejor orientados posible ya que esta posición será en la que la iluminación llegue con menos fuerza.

Otro inconveniente ocurre en la posición más cercana ya que el LED colocado al lado contrario al ojo que se utilice para realizar las pruebas provoca una reflexión que en el caso de algunos usuarios se junta con un reflejo producido en la esclerótica y en consecuencia, Shakti percibe ambos como un mismo punto superando los límites de área máxima del reflejo. No obstante, aunque Shakti no tuviese esta limitación del área, el percibir ambos reflejos como el mismo perjudicaría el funcionamiento del sistema. En consecuencia, se busca una solución a este problema otra vez jugando con la posición de los LEDs. Esta vez la solución consiste en acercar a la pantalla el LED del lado contrario al ojo utilizado para el seguimiento, posicionando el otro lo suficientemente alejado a esta como para que en la posición más alejada la distancia entre reflejos sea suficiente.

En conclusión, habrá que llegar a una posición de compromiso para que Shakti no pierda el seguimiento del ojo en ninguna de las posiciones.

### 6.3 Colocación del resto de elementos

Por último, habrá que concretar las distancias relativas entre la cámara, la pantalla y la orientación de la cámara, etc. (Ver fig. 6.2). La considerada posición central en este estudio se encuentra a 20,5cm de la mesa es decir, a 55cm de la cámara en horizontal, en línea recta. Sin embargo, el reposa cabezas que utilizaremos estará 10cm por encima de la mesa. La distancia entre el ojo del usuario y la cámara será de unos 60cm en la posición central.

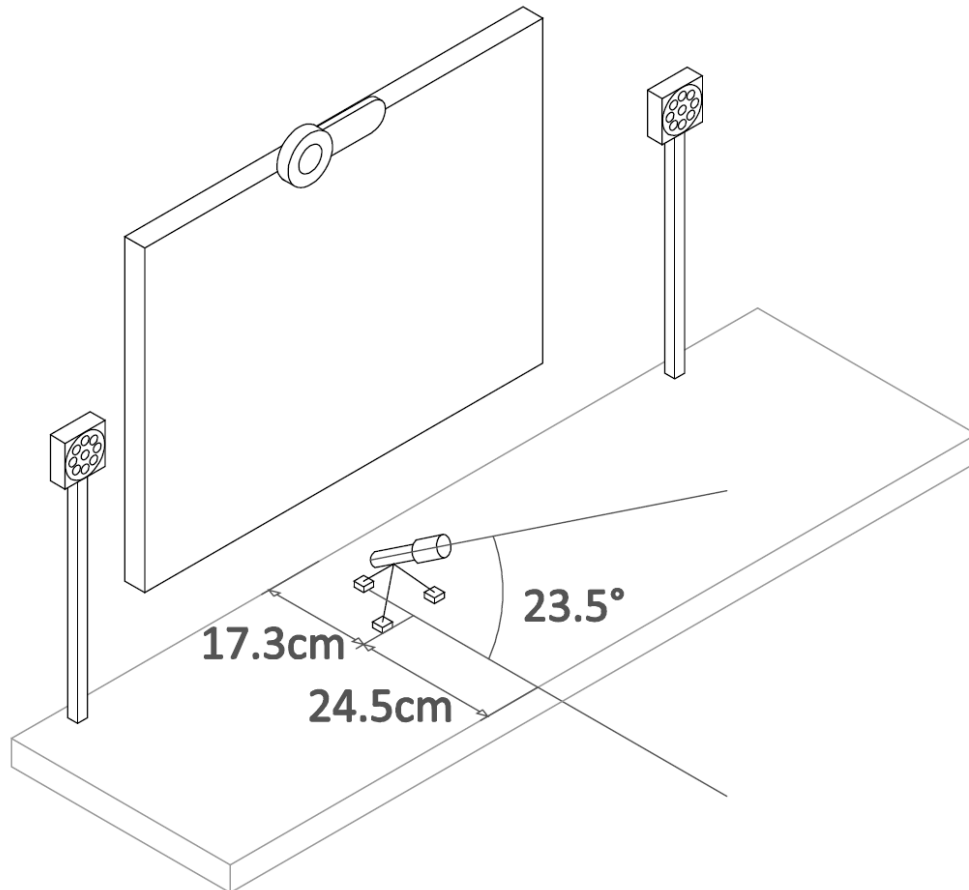


Fig. 6.2 Colocación de los elementos del sistema

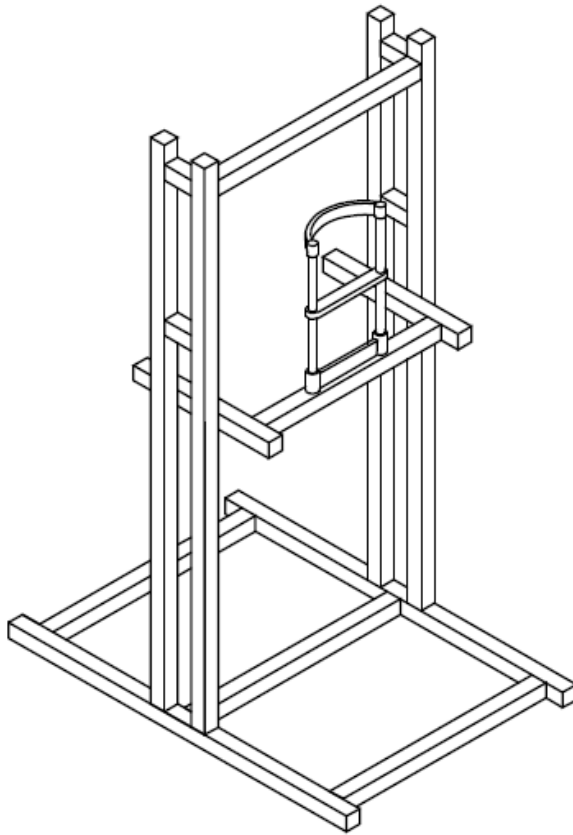
### 6.4 Descripción de las pruebas

Estas pruebas van a consistir en recorrer el eje Z realizando calibraciones en varios puntos, intentando permanecer en la misma posición en el eje X e Y. Al mismo tiempo que se realizan las calibraciones FaceAPI almacena los datos de la posición de la cabeza.

Primero, se colocará el usuario en la posición central y se enfocará la cámara para esa distancia manteniéndola invariante para las pruebas de ese usuario. Más tarde, se irán colocando los LEDs hasta conseguir que el seguimiento no se pierda en las posiciones más cercana y más alejada.

Una vez que el sistema garantice un correcto funcionamiento para dicho usuario se podrán empezar las pruebas. Para ello, empezando por la posición más cercana nos aseguraremos de que FaceAPI esté realizando el seguimiento de la cabeza del usuario y se procederá a realizar la primera calibración. Este proceso se repetirá para todas las posiciones. Los siete archivos de texto con la información de los datos de cada calibración serán almacenados así como el archivo de texto con los datos calculados a partir de FaceAPI.

Es importante recordar que la cámara, la pantalla y las siete posiciones del chinrest en donde se realizará cada una de las calibraciones permanecerán fijas para todos los usuarios. Sin embargo, la posición de los LEDs así como los parámetros de seguimiento podrán ser modificados para cada uno de los usuarios, manteniéndolos fijos para todas las posiciones.



Para la realización de estas pruebas se va a construir una sofisticada estructura que permita controlar la posición del usuario. Esta estructura permitirá un desplazamiento más controlado del usuario y a su vez incorpora un reposa cabezas que mantendrá a los usuarios en la misma posición de la cabeza en cada calibración (ver fig.6.3).

**Fig. 6.3** Estructura del reposa cabezas

#### 6.4.1 Protocolo experimental

En este apartado presentamos un guión del protocolo que hemos seguido para la realización del estudio experimental. Para estas pruebas se ha decido realizar el seguimiento del **ojo derecho**.

- 1.- Usuario situado en la posición 4.
- 2.- Se enfocará la cámara.
- 3.- Usuario situado en la posición más cercana al sistema.
- 4.- Ajuste de LED izquierdo para que la reflexión producida por la esclerótica no perjudique la detección del correspondiente reflejo de la córnea.
- 5.- Usuario situado en la posición más alejada al sistema.
- 6.- Se orientan los LEDs hacia el usuario para que la iluminación sea suficiente como para detectar los reflejos de la córnea. También hay que vigilar que la distancia entre los reflejos sea la adecuada y sino alejar el LED derecho.
- 8.- Nuevamente el usuario situado en la posición más cercana al sistema.
- 9.- Comprobar que el cambio de posición de los LEDs no provoque la pérdida del seguimiento en esta posición y sino repetir los pasos del 3 al 8 hasta conseguir que el seguimiento no se pierda en todo el rango. Entonces podrán empezar las pruebas.

- 10.-Lanzamos FaceAPI y esperamos hasta que empiece el seguimiento de la cara y activaremos el guardado de los datos.
- 11.-Realizamos una calibración en Shakti y almacenamos el archivo de texto de la calibración.
- 12.-El usuario avanza a la siguiente posición (2cm en el eje Z).
- 13.-Esperamos a que FaceAPI reanude el seguimiento para realizar la siguiente calibración (que será posteriormente almacenada).
- 14.-Se repite el mismo proceso para las siete posiciones.
- 15.-Finalmente se almacenará el archivo de texto creado por FaceAPI con la información de la posición y rotación de la cabeza y la posición del ojo.

## 6.5 Sincronización entre Shakti y FaceAPI

Para corregir la ecuación de calibración de partida, queremos incorporar al sistema de seguimiento de la mirada información de la posición de la cabeza conocida con la ayuda de FaceAPI. Con la ayuda de Matlab se va a sincronizar Shakti con FaceAPI. La información relativa a los seguimientos tanto de la cara con FaceAPI como de la mirada con Shakti es extraída a unos archivos de texto. Ambos archivos incluyen información del tiempo en el que se ha realizado cada una de las capturas.

Para realizar la sincronización extraeremos la información del tiempo de las capturas de ambos programas. Tomando la información de Shakti de partida, se calcularán las dos capturas de FaceAPI más próximas en el tiempo a cada captura de Shakti. Para establecer un único valor de la posición de la cabeza de las dos capturas más próximas en el tiempo de FaceAPI se realizará una media ponderada. Es decir, se calculará la media de ambas capturas ponderada por la diferencia existente entre el tiempo de cada una de ellas al tiempo de la captura considerado por Shakti (ver figura 6.4).

tiempo

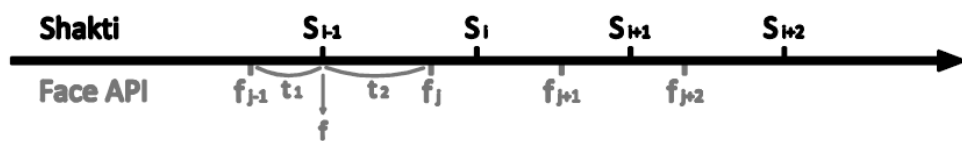


Fig. 6.4 Sincronización de FaceAPI con Shakti.

$$f = \frac{(f_{j-1} * t_2) + (f_j * t_1)}{(t_1 + t_2)} \quad (6.1)$$

Es decir para la muestra  $S_{i-1}$  de Shakti la información de FaceAPI obtenida será corresponderá con  $f$  (ecuación 6.1). Donde  $f_{j-1}$  y  $f_j$  serán las muestras más cercanas de FaceAPI en el tiempo de la captura  $S_{i-1}$  de Shakti.

## 6.6 Conclusiones

El objetivo de estas pruebas será caracterizar el error producido por la ecuación de calibración de partida (ecuaciones 3.1 y 3.2) frente a desplazamientos controlados, y así, conociendo la tendencia del error de cada movimiento, corregir esta ecuación con el objetivo de minimizar este error proporcionando una mayor libertad de movimientos al usuario sin perder precisión. Habiendo considerado diferentes opciones de movimientos a estudiar se decide estudiar el efecto del sistema ante desplazamientos en el eje Z del usuario ya que como estudios anteriores advierten [9] y [19] es el movimiento más crítico.

Concretamente se van a estudiar 7 posiciones diferentes distanciadas 2cm entre sí. La cuarta posición será la posición donde será enfocada la cámara. Por tanto, estas pruebas pretenden comparar el error de estimación cometido en el resto de posiciones respecto a la calibración realizada en la posición central. Será necesario el uso de un reposa cabezas.

Antes de realizar las pruebas hay que conseguir que el sistema no pierda el seguimiento del ojo de cada usuario en todas las posiciones consideradas. Para ello han sido necesarios unos cambios en el sistema:

- Se ha cambiado el objetivo de 35mm de la cámara por uno de 16mm para no perder el ojo de los usuarios del plano imagen en todas las posiciones sin mover la cámara para ello.
- La posición de los iluminadores pasa a construir un grado de libertad más del sistema permitiendo su ajuste para cada usuario. Ya sea por la separación mínima requerida entre los reflejos, la orientación de los mismos para una iluminación suficiente en las posiciones más alejadas o para evitar las reflexiones producidas por la esclerótica.
- Así como la posición de los iluminadores, los parámetros de seguimiento también podrán ser ajustados por usuario.

No obstante, la posición de la cámara, del reposa cabezas (para cada posición) y la pantalla deberán permanecer en el mismo lugar para todos los usuarios.

En este estudio participarán un total de doce usuarios para medir el error cometido por la ecuación de partida ante desplazamientos de los usuarios. En comparación con estudios anteriores [3], [4], [6], [9], el número de usuarios es muy elevado, dotando al estudio de gran relevancia práctica.

## 7 Análisis de los resultados

En el presente apartado se van a estudiar los resultados de las pruebas descritas en el apartado anterior. A partir de estos resultados intentaremos corregir la ecuación de calibración de partida para optimizar su comportamiento permitiendo así libertad de movimientos al usuario.

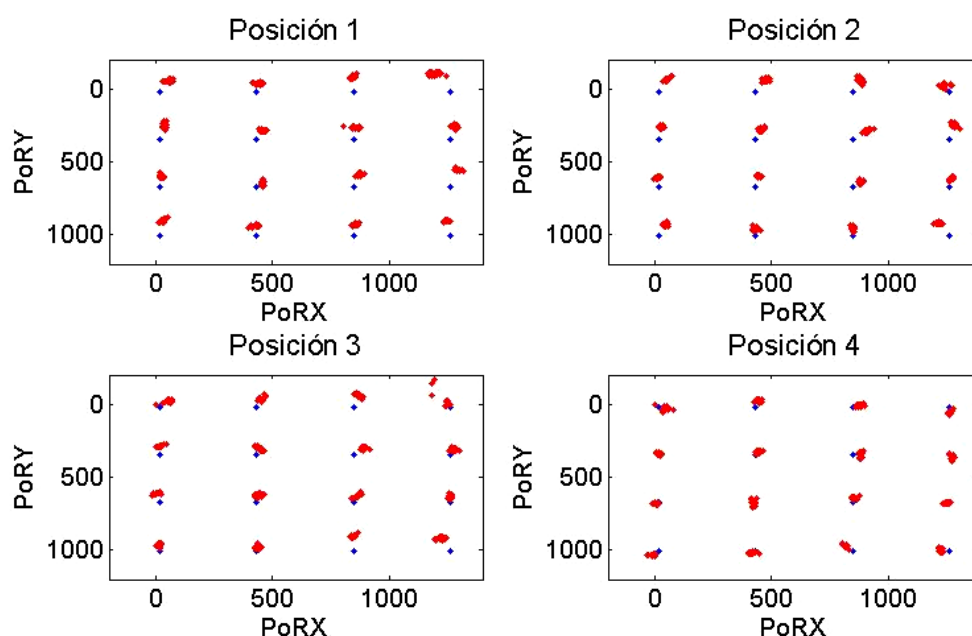
Para caracterizar el error de estimación de cada una de las posiciones consideradas para realizar las pruebas, tomamos la posición central de referencia y las demás serán analizadas respecto a esta. Por ello, hemos calculado con los datos de Shakti de la posición central de cada usuario su correspondiente ecuación de calibración. Los coeficientes de esta ecuación serán utilizados para la estimación del  $PoR$  del resto de las ecuaciones.

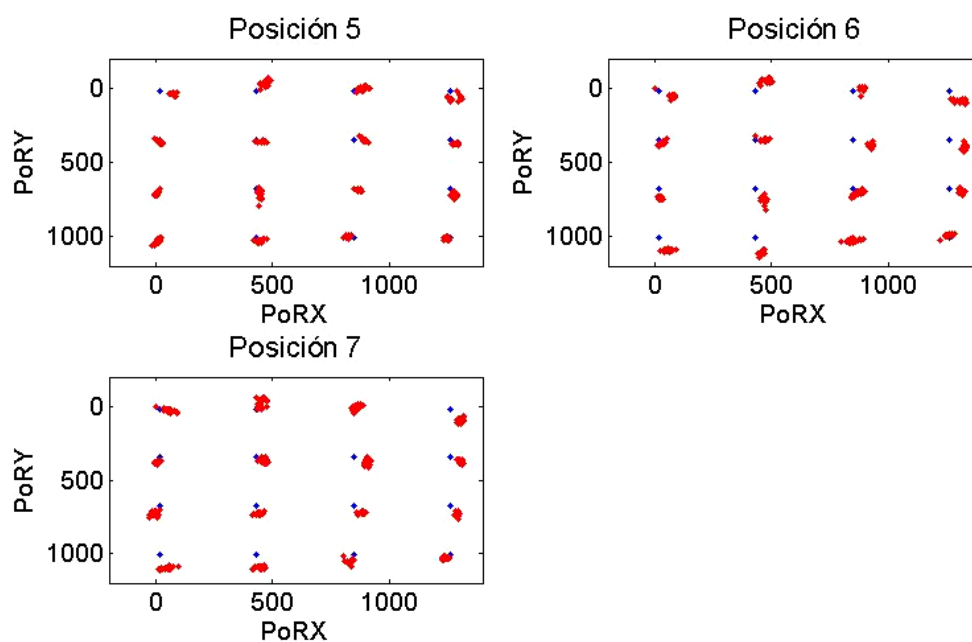
A continuación, se representan el error que presenta la estimación del  $PoR$  a través de la ecuación de calibración respecto al  $PoR$  real en las distintas posiciones en las que los doce usuarios han calibrado.

### 7.1 Presentación de los resultados

#### 7.1.1 Cálculo del $PoR$ para cada posición

Una forma de poder visualizar la evolución de la estimación del  $PoR$  o el comportamiento del sistema a medida que el usuario se desplaza por el eje  $Z$  sería mediante una secuencia de fotogramas donde cada posición corresponda con un frame. Con esta misma idea observaremos la estimación del  $PoR$  de todas las capturas (filtradas) en cada una de las fijaciones de todas las posiciones respecto al  $PoR$  real. Donde se muestren en rojo la estimación del  $PoR$  calculado por la ecuación de calibración y el  $PoR$  real conocido por Shakti en azul. Aquí se representa cada uno de los frames, que corresponde con cada una de las posiciones, en una figura independiente. Solo se va a mostrar la información del usuario 12 ya que la mayoría de los usuarios presentan un comportamiento parecido a este.

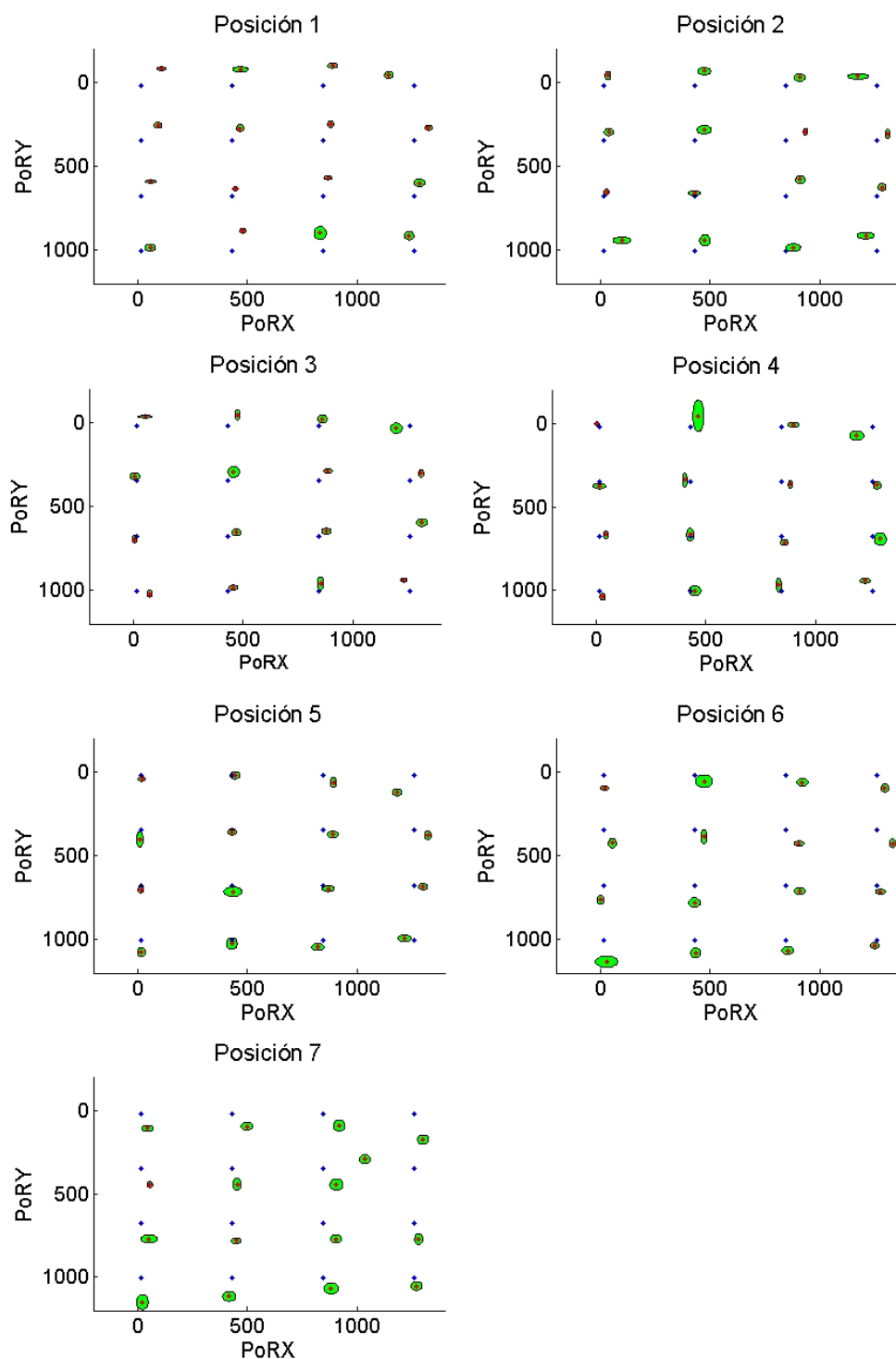




**Fig. 7.1** Estimación del  $PoR$  mediante la ecuación de calibración. Usuario 12

Los 7 gráficos de la figura anterior (fig. 7.1) muestran la evolución de la estimación del  $PoR$  en función de la posición de calibración. Claramente se observa como el  $PoR$  estimado es más próximo al  $PoR$  real en la posición central, es decir, la posición 4. Además conforme avanzamos de la posición más cercana a la cámara (Posición 1) a la más lejana (Posición 7), la estimación de  $PoRY$  aumenta.

Además de este modo de representación en el que se observan todas las imágenes de cada una de las fijaciones para cada posición, otra forma de representar estos datos es representar la media de la estimación del  $PoR$  en cada fijación y la correspondiente desviación estándar. Siguiendo el mismo esquema de la figura anterior (fig. 7.1), en esta figura (fig. 7.2) se representa en azul el  $PoR$  real, en rojo la media de la estimación del  $PoR$  en cada fijación, y en verde la desviación estándar de las imágenes de cada fijación. Para este siguiente ejemplo mostraremos la estimación del  $PoR$  del usuario 10. Así quedará comprobado que el comportamiento de todos los usuarios es similar.



**Fig. 7.2** Estimación del PoR medio y desviación estándar. Usuario10

La desviación de cada usuario es muy baja en todas las posiciones aunque en general esta aumenta para las posiciones más alejadas. En general, se observa la misma tendencia que en la figura anterior. Todos los usuarios presentan un comportamiento parecido a este, aunque en algunos casos la desviación de la posición más alejada y más próxima a la cámara es mayor.



### 7.1.2 Trayectoria del *PoR* estimado

Para englobar todas las posiciones de un usuario en un mismo gráfico se muestra la trayectoria del *PoR* medio de cada fijación calculado independientemente para cada usuario. En este caso, en vez de representar todas las imágenes (filtradas), solo se representará la media de cada una de las fijaciones y todas las posiciones simultáneamente unidas por una línea desde la posición más cercana a la más lejana. Esta representación nos indica de forma directa la tendencia que siguen los datos ante el desplazamiento del usuario. La primera posición está representada mediante el punto verde y el azul indica el *PoR* real.

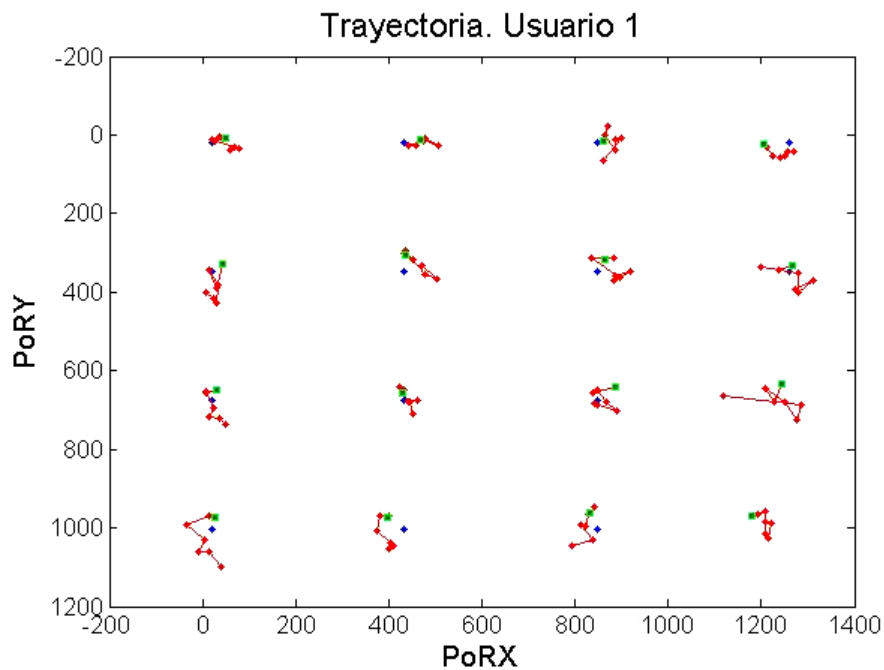


Fig. 7.3 Trayectoria del *PoR* medio. Usuario1

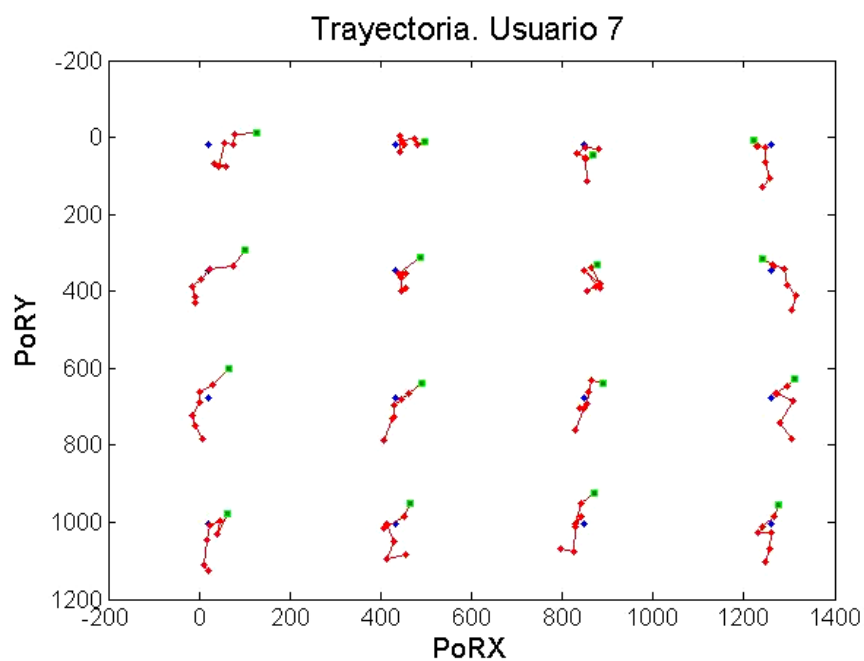
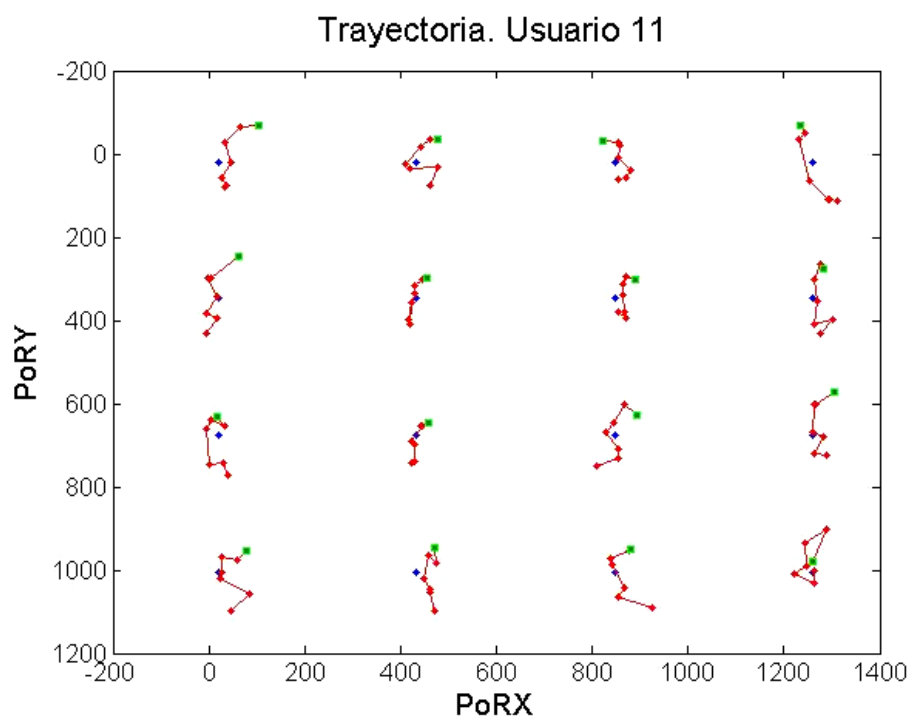
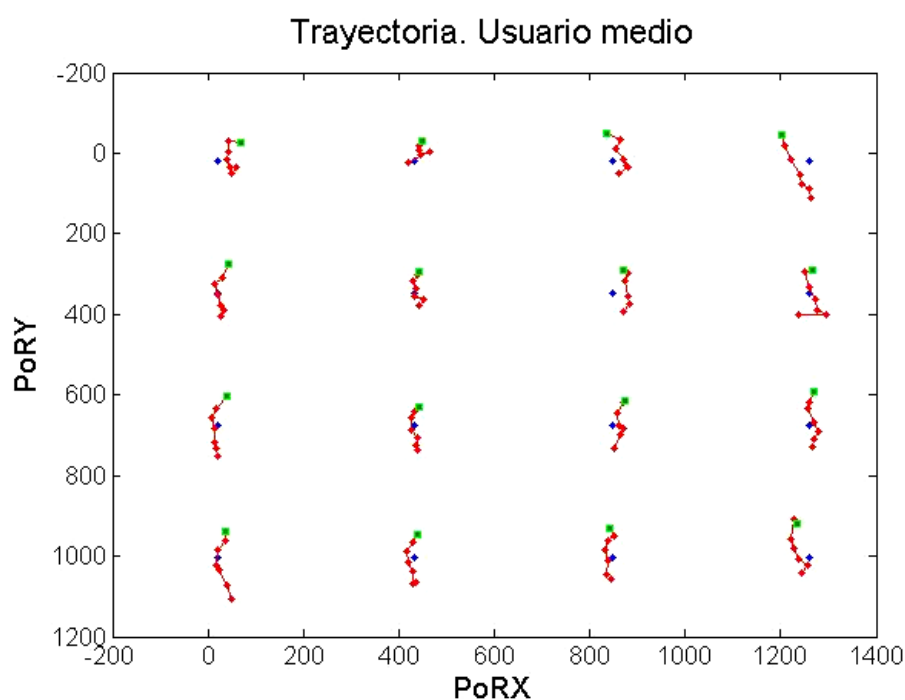


Fig. 7.4 Trayectoria del *PoR* medio. Usuario7



**Fig. 7.5** Trayectoria del PoR medio. Usuario11



**Fig. 7.6** Trayectoria del PoR medio. Usuario medio

Los tres primeros gráficos (fig. 7.3, 7.4 y 7.5) muestran la trayectoria del *PoR* medio en todas las fijaciones y todas las posiciones de tres de los usuarios de los 12 que realizaron las pruebas. Y en el último gráfico se muestra la trayectoria media de todos los usuarios. Claramente se observa que el error en el eje Y es más significativo que el error en X, en el primer caso en las posiciones más alejadas llega a ser de cerca de 70 píxeles mientras que el

error en X no supera los 30 píxeles. En todos ellos se nota una tendencia del *PoR* que aumenta en el eje Y de forma lineal conforme se avanza a las posiciones más alejadas en el eje Z, además, al promediar la tendencia se hace todavía más evidente (fig.7.6). Esta tendencia aparente, es decir, la linealidad del error en el eje Y con la distancia en Z del usuario resultará crucial a la hora de optimizar la ecuación ya que lo que se buscaba con estas pruebas es estudiar si existe alguna relación con el movimiento en el eje Z y el error cometido por la ecuación de calibración. Obviamente este resultado muestra la clara dependencia entre el error en el eje Y y el desplazamiento del usuario en el eje Z. Con esto, se intentará corregir la ecuación de calibración (únicamente *PoRY*) ante este tipo de desplazamientos.

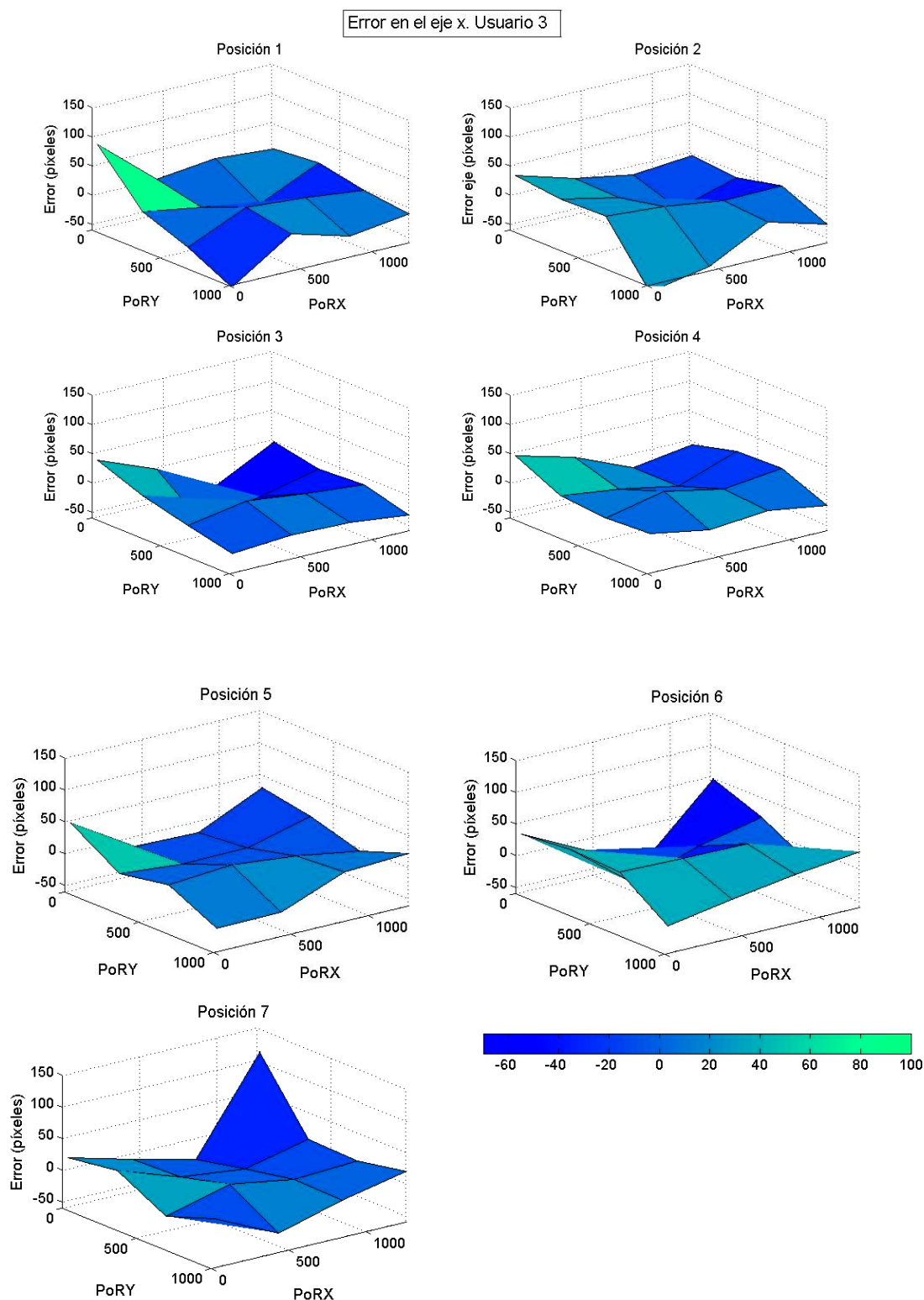
### 7.1.3 Error entre el *PoR* real y el estimado

A partir de la estimación del *PoR* para cada una de las fijaciones de una posición, podemos también calcular el error por fijación como la media de las diferencias entre el *PoR* real y el *PoR* estimado de cada imagen. Por cada posición por tanto, obtendremos 16 valores de error, uno por fijación. Para realizar el cálculo del error se considerarán el error en el eje X y el error en el eje Y independientemente ya que la ecuación de calibración es diferente para cada una de las coordenadas. También puede calcularse el error absoluto siguiendo la siguiente fórmula:

$$\text{Error\_absoluto}_{i,j,k,z} = \sqrt{((\text{error\_x}_{i,j,k,z})^2 + (\text{error\_y}_{i,j,k,z})^2)}$$

El error absoluto del usuario i-ésimo, en la posición j-ésima, de la fijación k-ésima y observación z-ésima. A partir del error de cada imagen se hará un promedio de todas las imágenes de cada fijación y para cada posición de calibración de cada usuario.

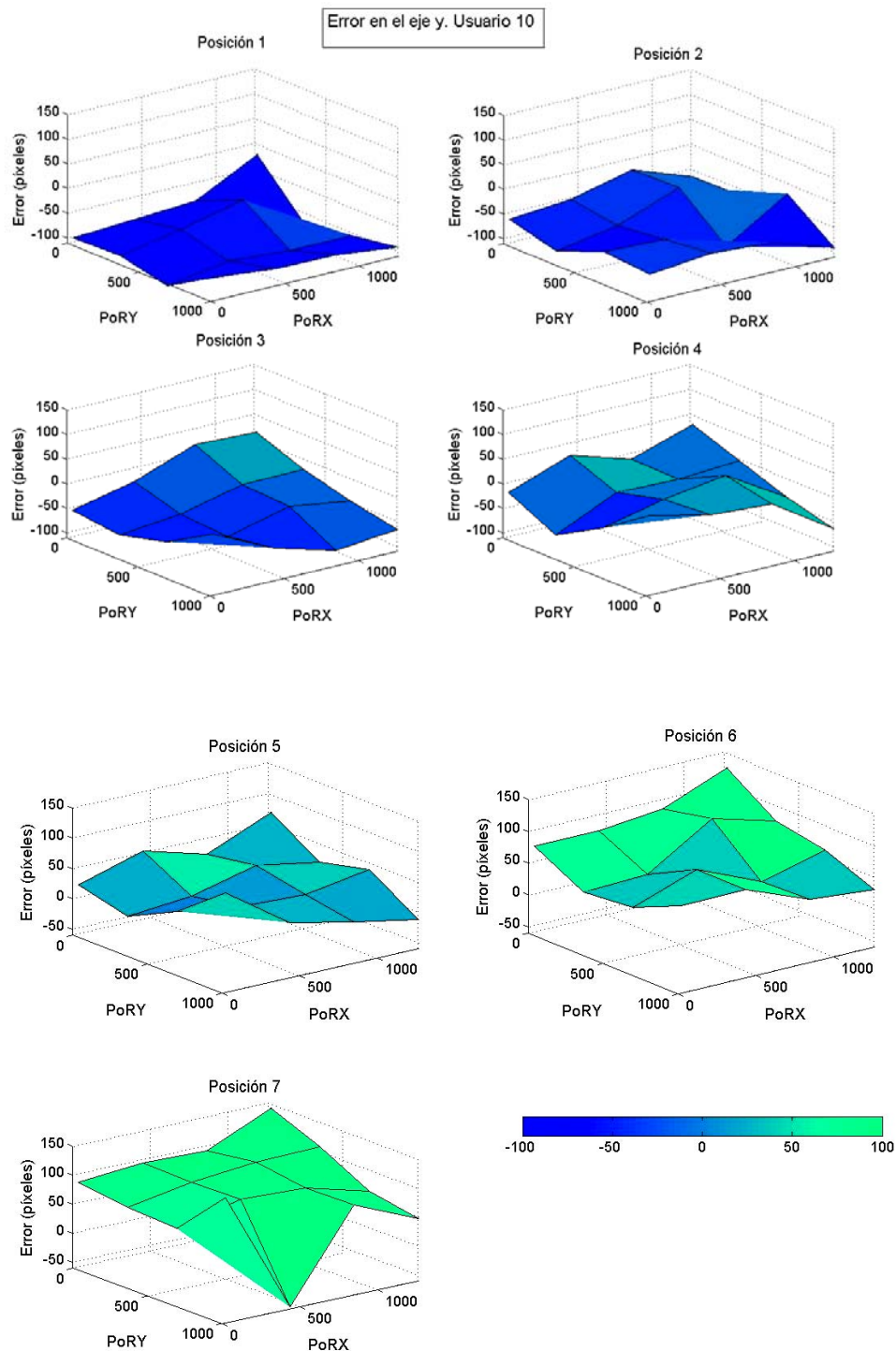
La representación del error cometido por la estimación del *PoRX* por la ecuación de calibración (ver fig. 7.7) no es muy ilustrativa ya que no se observa ninguna tendencia aparente al avanzar de posición. Además, en la figura 7.7 ya deducimos que la principal desviación entre la estimación del *PoR* y el *PoR* real se aprecia en el eje Y, donde observamos una dependencia clara con el movimiento en el eje Z. Para evaluar la cuantía del error tomaremos como referencia un estudio anterior de la ecuación de partida que utilizamos [1]. El error máximo (absoluto) calculado con esta ecuación en la posición de calibración es de unos 35 píxeles lo cual coincide con el error de la posición 4. A partir de estos datos podemos evaluar el incremento del error en otras posiciones. Sin embargo, este decremento no es muy significativo exceptuando las posiciones más alejadas de la cámara en las que el error aumenta en algunos puntos de la pantalla, en las fijaciones extremas de la pantalla. En las posiciones más alejadas el sistema puede fallar ya que el desenfoque de la cámara es mayor y la iluminación de los LEDs también, lo cual puede afectar a la detección de features. Tampoco se observa un comportamiento constante para todas las posiciones, solamente que el error es mayor en algunas de las esquinas de la pantalla.



**Fig. 7.7** Error de la estimación del  $PoR$  medio en el eje  $Y$ . Usuario3

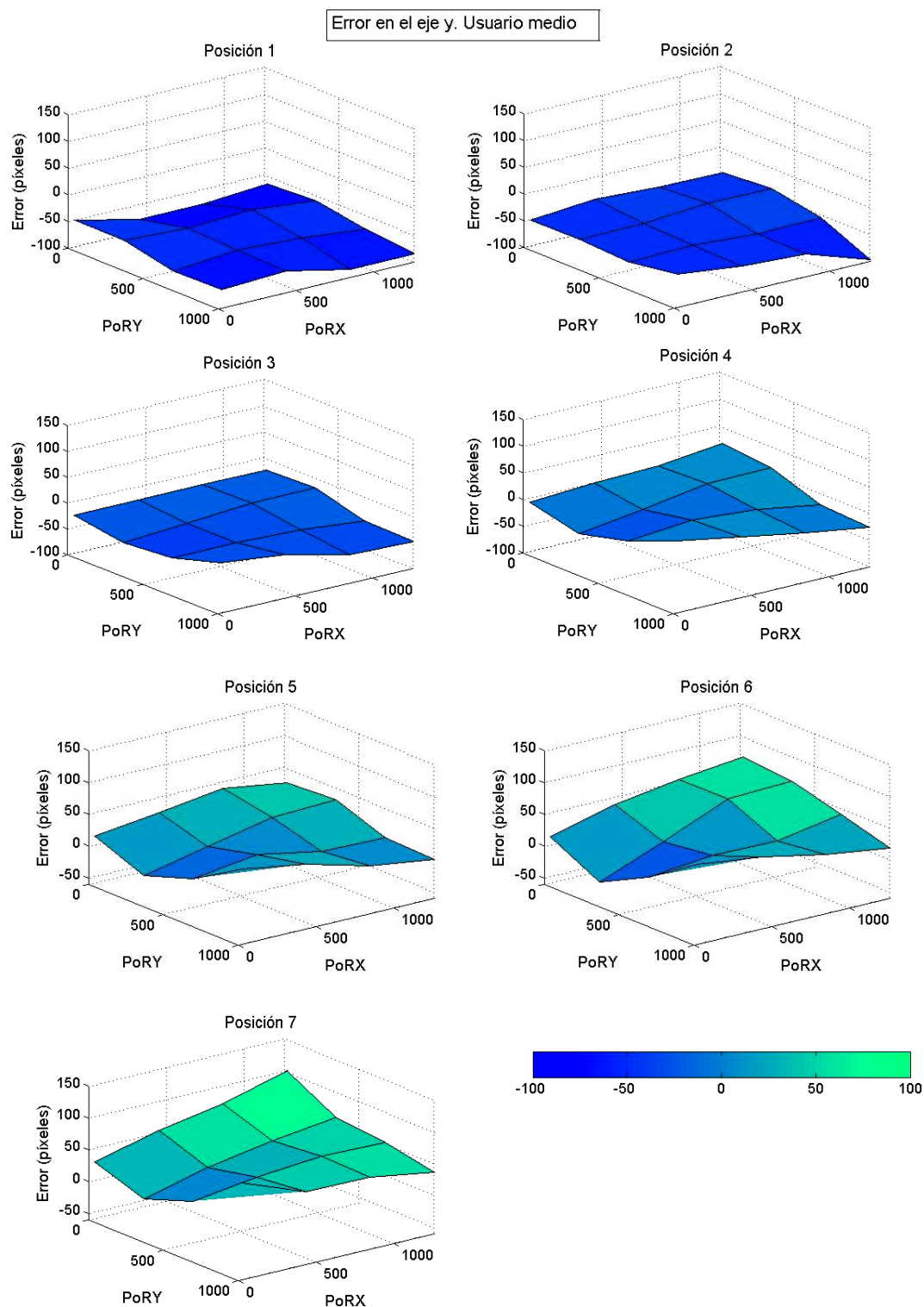
En la siguiente figura (fig. 7.8) de la misma manera que la anterior se representará el error de la estimación del  $PoRY$  respecto al  $PoRY$  real del usuario 10. Como era de esperar después de haber observado la trayectoria del  $PoR$  al avanzar de posición (ver fig. 7.6), el error en el eje  $Y$  está entorno a cero en la posición central (4), al avanzar hacia las posiciones más cercana a la

cámara el error va aumentando (aunque es negativo) y al avanzar hacia las posiciones más alejadas a la cámara el error también aumenta.



**Fig. 7.8** Error de la estimación del PoR medio en el eje Y. Usuario 10

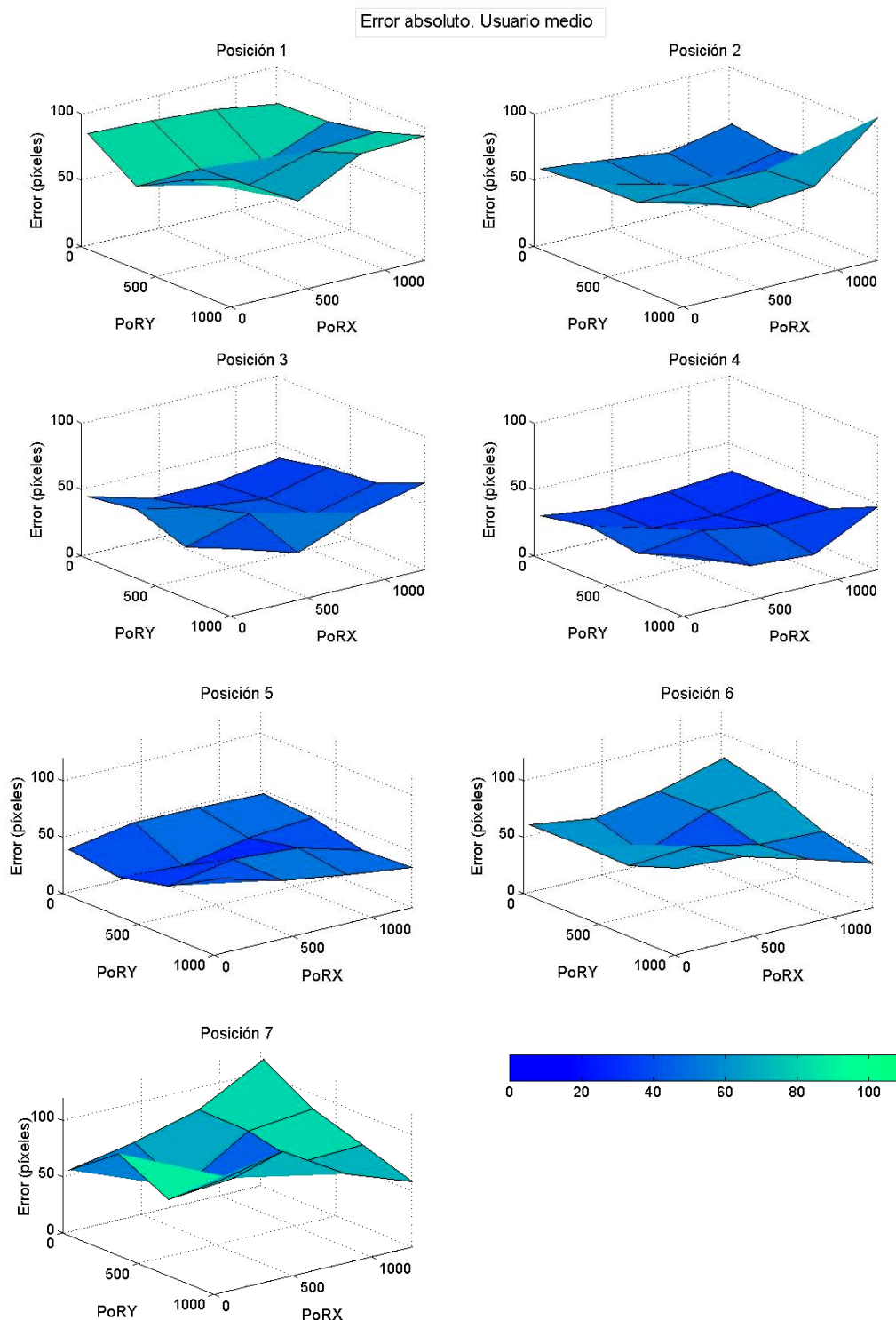
A continuación, se muestra el error en el eje  $Y$  de la misma manera que en la figura 7.8 del usuario medio:



**Fig. 7.9** Error de la estimación del  $PoR$  medio en el eje  $Y$ . Usuario medio

En este caso, la tendencia mostrada por el usuario 10 o cualquier otro que no haya sido representado se hace mucho más evidente (fig 7.9). Es decir, se observa una clara dependencia entre la posición en el eje  $Z$  y el error en el eje  $Y$  cometido al estimar el  $PoR$ . En este caso el error se mantiene constante para todas las fijaciones de la misma posición, desapareciendo los errores puntuales que pueda causar cada usuario en concreto.

Por último se va a representar el error absoluto del usuario medio (ver fig. 7.10):



**Fig. 7.10** Error de la estimación del  $PoR$  medio absoluto. Usuario medio

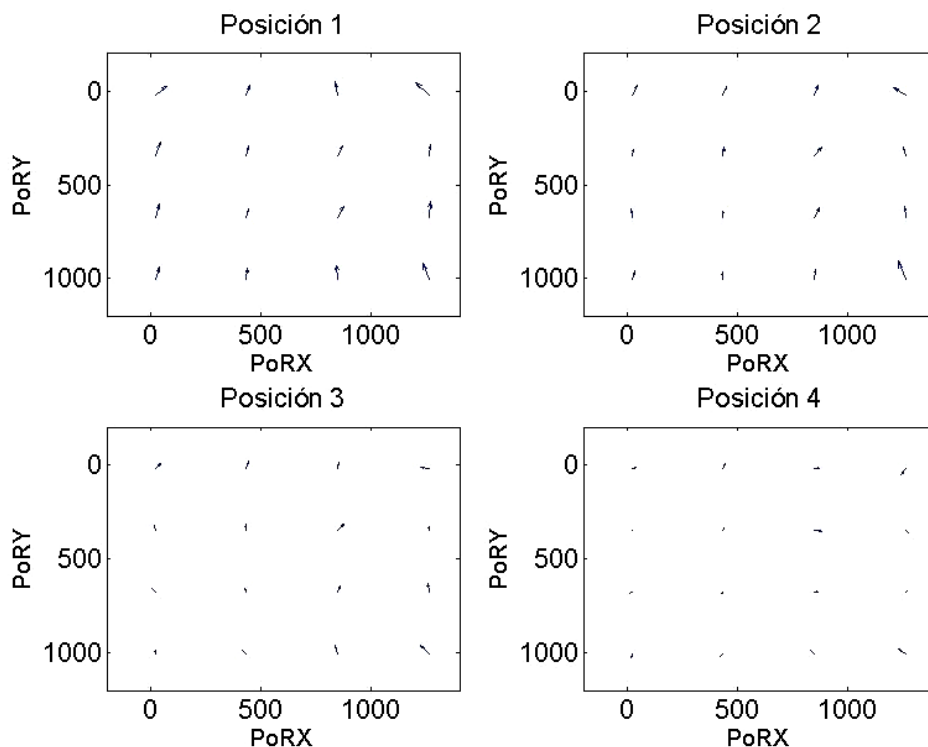


El error absoluto aumenta conforme el usuario se aleja de la posición central, la cuarta (ver fig. 7.10). Este efecto es provocado a causa del error en el eje  $Y$  producido por los desplazamientos del usuario en el eje  $Z$ . Como ya se predecía al representar la estimación del  $PoR$ , calculando el error se confirma la dependencia del aumento del error en el eje  $Y$  con el desplazamiento del usuario.

#### 7.1.4 Dirección del error

La última representación del error entre la estimación del  $PoR$  y el  $PoR$  real promediado por cada fijación se trata de un vector que muestra la dirección del error en cada fijación, cuya longitud corresponde con el valor del error absoluto.

Esta representación muestra claramente como avanza la dirección del error absoluto conforme con el movimiento del usuario (ver fig. 7.11). De la posición central a la primera se observa como la estimación del  $PoR$  va alejándose del  $PoR$  real, es decir el error está aumentando. Además como ya veníamos comentando el error se observa principalmente en el eje  $Y$ , al acercarse a la cámara desde la posición central la estimación del  $PoRY$  es inferior respecto al  $PoRY$  real. Lo mismo ocurre al alejarnos de la cámara, el error aumenta, sin embargo, esta vez la estimación del  $PoRY$  es superior a la del  $PoRY$  real. Esta representación (fig. 7.11) tiene una directa relación con la figura 7.6 en la que se representa la trayectoria del error definida por el cambio de dirección de cada vector de una posición a otra.





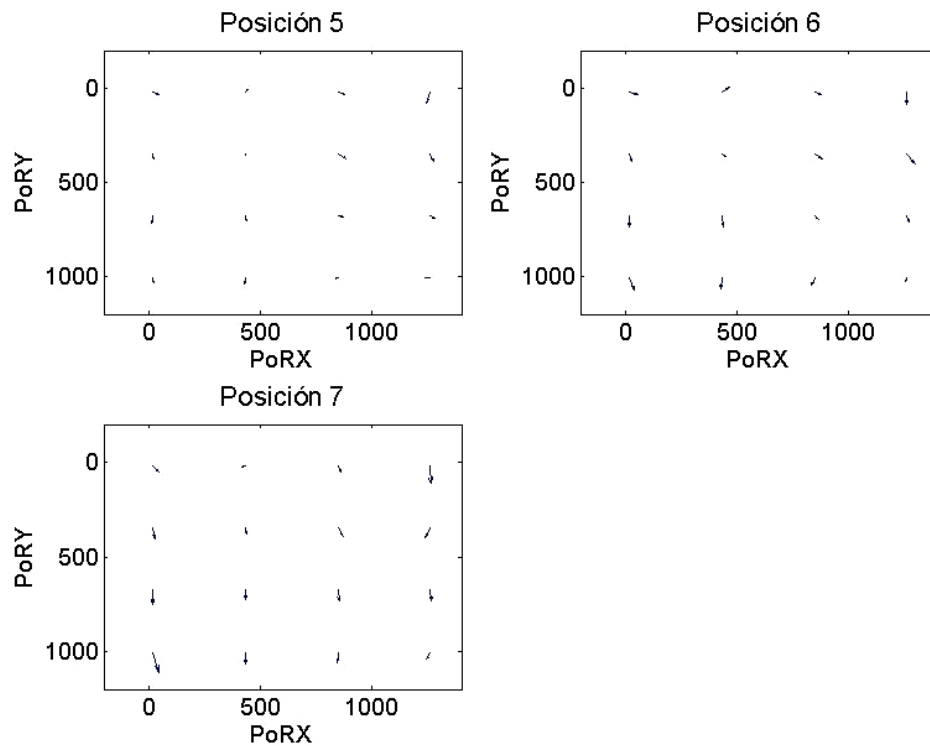


Fig.7.11 Vector de tendencia del error. Usuario medio

### 7.1.5 Comportamiento medio en cada posición.

En este apartado mostraremos el error medio de cada posición de todos los usuarios, promediando el error absoluto de todas las fijaciones, diferenciando el error en el eje X (fig. 7.12), el error en el eje Y (fig. 7.13) y el error absoluto por separado (fig. 7.14). En azul está representada la media del error por posición de cada uno de los usuarios independientemente y la línea roja es la media de todos los usuarios.

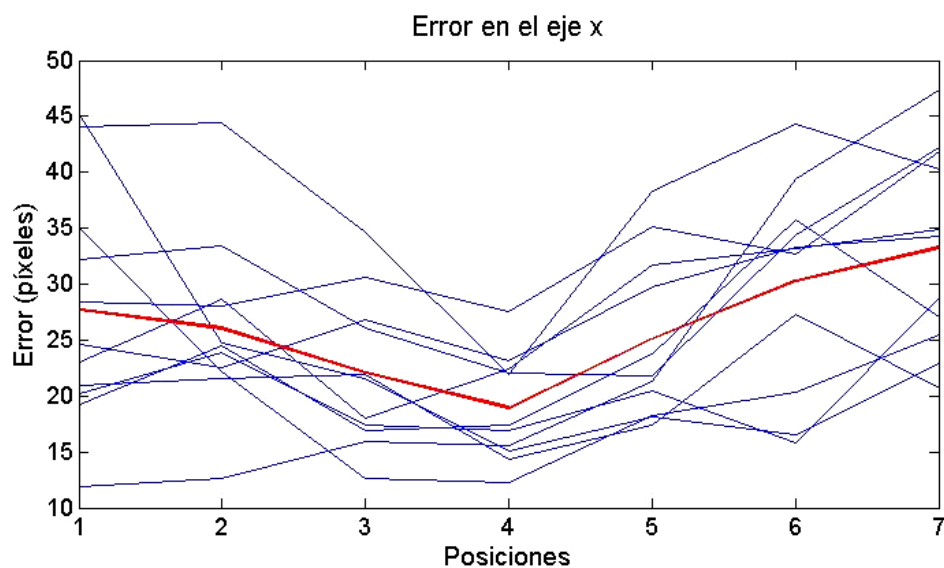


Fig.7.12 Error medio X de cada posición

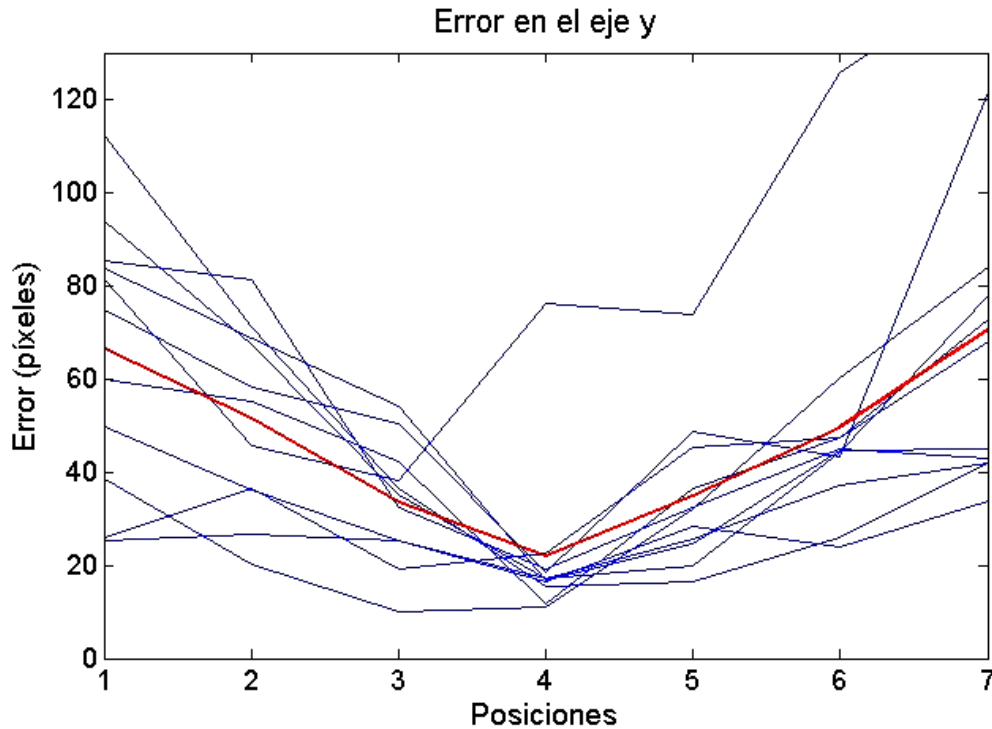


Fig.7.13 Error medio en el eje Y de cada posición

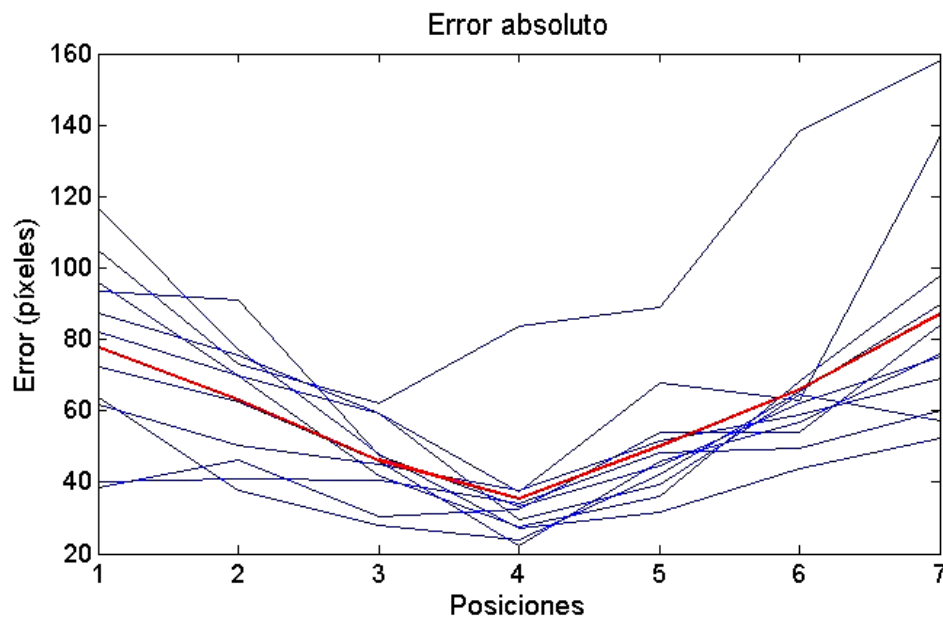
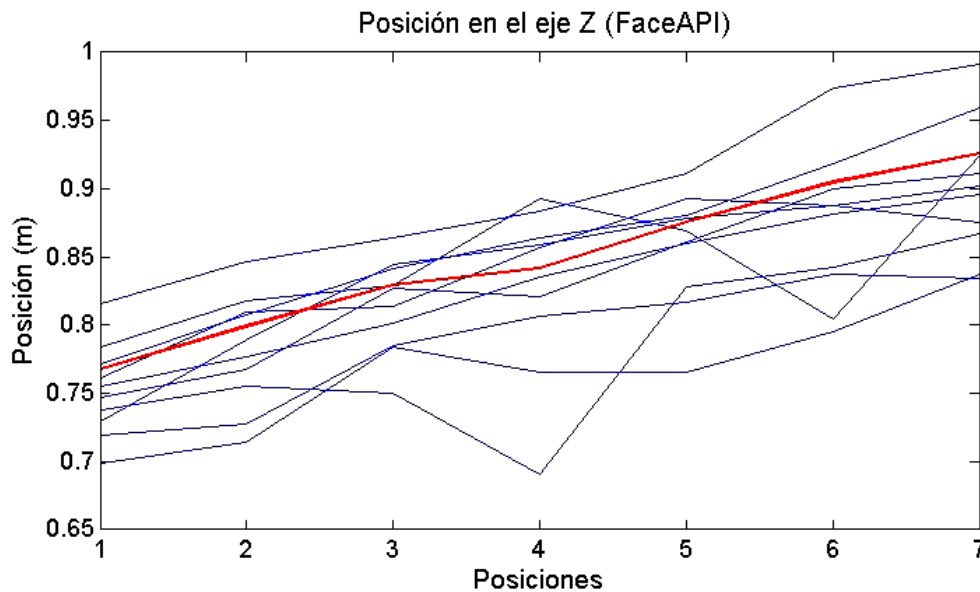


Fig.7.14 Error medio-absoluto de cada posición

Los gráficos anteriores (fig. 7.12, 7.13 y 7.14) describen el comportamiento del error conforme cada usuario avanza en el eje Z. Para el error medio en X aparentemente, se podría deducir cierta linealidad en el comportamiento del usuario medio (línea roja) con los desplazamientos del usuario. Pero, es de mucha menos relevancia que en el caso del error en Y, ya que la magnitud del error en X es mucho menor. (El rango del eje vertical de los tres gráficos no es el mismo). Además observando cada usuario por separado esta tendencia no es

tan clara aunque en general en la posición central el error es inferior. En cambio, el caso del error en el eje  $Y$  (ver fig. 7.13) muestra cierta similitud entre usuarios ya que el error aumenta progresivamente conforme nos alejamos de la posición central a posiciones más lejanas y a las posiciones más cercanas al sistema. El error absoluto (fig. 7.14) sigue el mismo comportamiento debido a la componente  $Y$  del error.

A continuación incluimos la información de FaceAPI de la posición en el eje  $Z$  de todos los usuarios (fig 7.15) ya que para poder corregir la ecuación de calibración ante este desplazamiento podría ser una fuente valiosa de información.



**Fig.7.15** Posición en el eje  $Z$  (FaceAPI) media por cada posición

La información de la posición de la cabeza media ofrecida por FaceAPI (fig. 7.15) mantiene un comportamiento casi lineal con las posiciones en las que se ha calibrado. Sin embargo, este comportamiento no es tan claro para cada uno de los usuarios estudiándolos por separado, lo cual podría perjudicar la ecuación de calibración en caso de que se utilizase esta información como corrección. Además, hemos comprobado que el error de estimación del  $PoR$  aumenta considerablemente al desplazarnos de una posición a la siguiente (2cm), sin embargo, FaceAPI comete un error de 1cm en el cálculo de la posición del usuario. Es decir, la precisión de Shakti es mucho mayor que la de FaceAPI. Con esto, pondremos en duda el comportamiento de FaceAPI para corregir la ecuación de calibración.

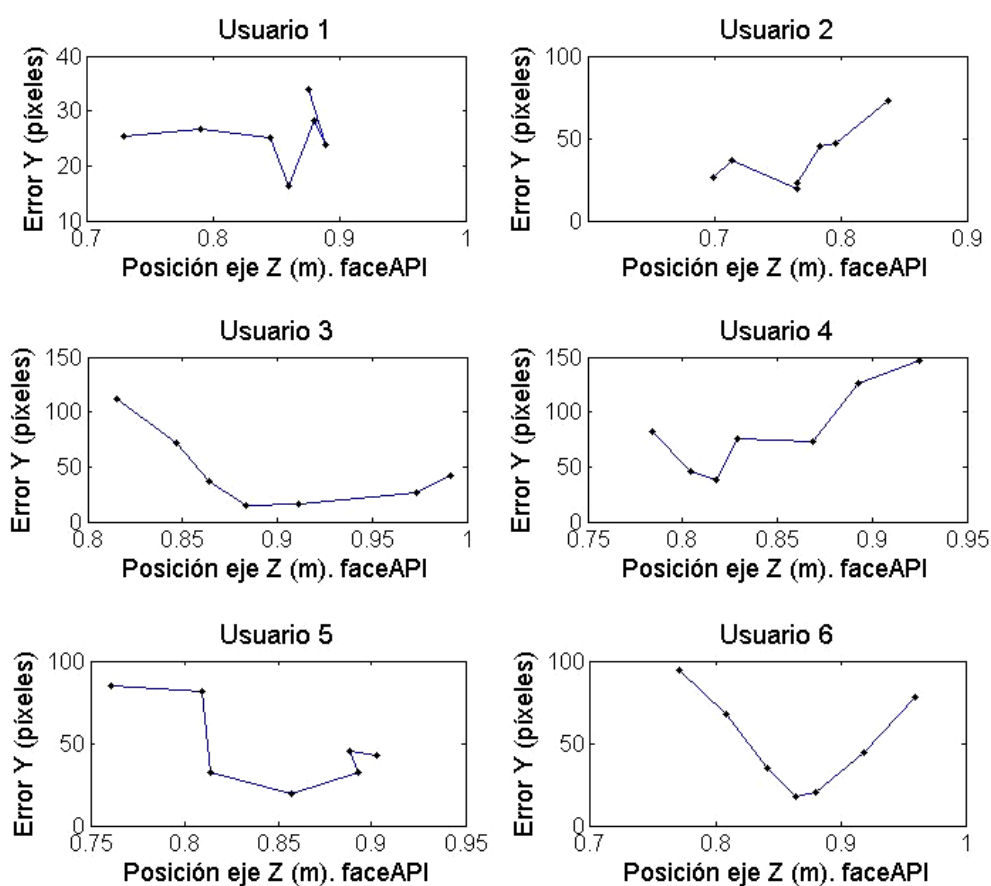
### 7.1.6 Correlación entre FaceAPI y el error en $Y$ .

Con el fin de optimizar el sistema corrigiendo el error cometido al estimar el  $PoRY$  ante desplazamientos en el eje  $Z$ , se pondrá en relación dicho error con la información de la posición en  $Z$  estimada por FaceAPI.

Para ello vamos a representar el error en el eje  $Y$  medio de las siete posiciones frente a la posición en  $Z$  correspondiente calculada a partir de FaceAPI. Se representará cada usuario en una gráfica diferente. (Ver figura 7.16).

En las siguientes gráficas (fig 7.16) podemos ver que el incremento de la posición en Z calculada por FaceAPI no es lineal con la posición real. Es decir, en las pruebas avanzamos 2cm entre cada una de las posiciones sin embargo, observando el resultado de FaceAPI parece que esta distancia no fuese constante. En algunos casos incluso la posición calculada por este sistema es menor a la posición inmediatamente anterior. Además como ya veníamos diciendo en la caracterización de FaceAPI los datos de la posición en Z de los diferentes usuarios no coinciden a pesar de que las posiciones reales coincidiesen en las pruebas. En cuanto al error cometido al estimar el *PoRY* observamos que este aumenta a medida que nos separamos de la posición central, la cuarta.

El objetivo de este estudio es poner en relación la posición de la cabeza en Z calculada por FaceAPI con el error al estimar el *PoRY*. Así podríamos corregir la ecuación de calibración añadiendo un término de corrección al determinar la posición de la cabeza en el eje Z. Pero, a la vista de estos resultados, al observar la imprecisión de los cálculos de FaceAPI buscaremos otras opciones de optimización con diferentes variables que nos proporcionen información sobre la posición en Z del usuario.



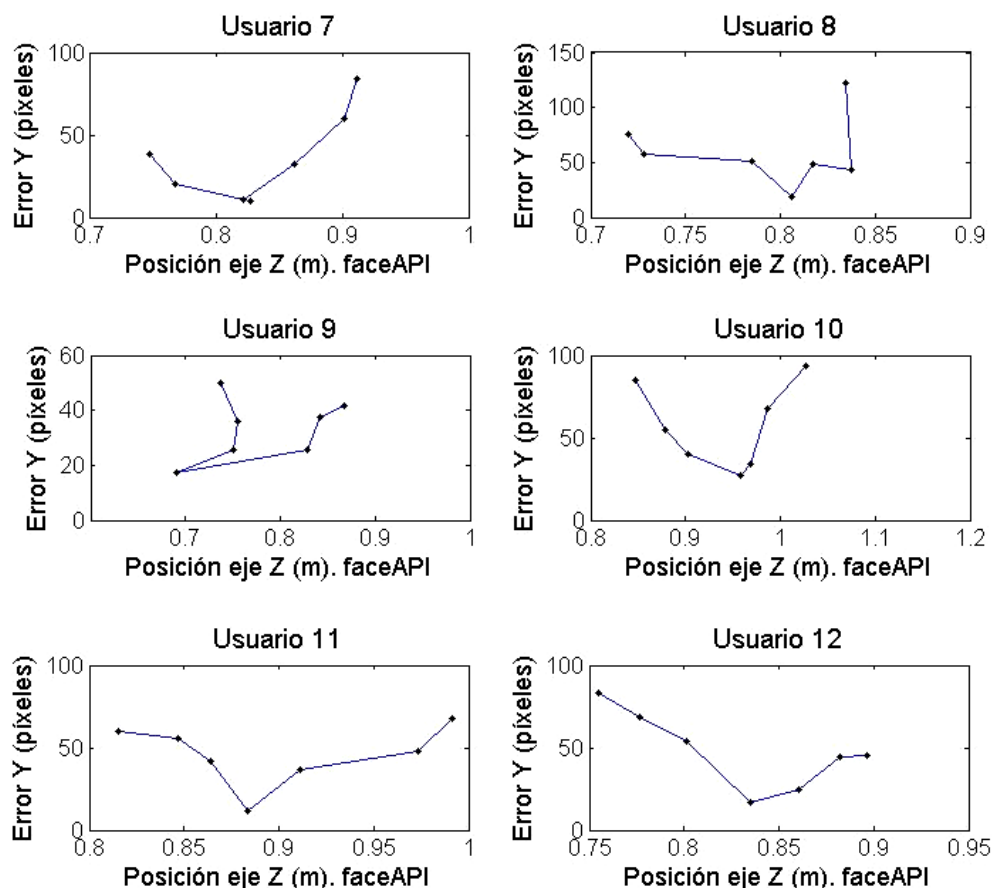


Fig.7.16 Error en el eje Y frente a la posición en el eje Z calculada por FaceAPI

### 7.1.7 Conclusiones

En todas las representaciones posibles de los resultados de las pruebas que han sido mostradas en los apartados anteriores se percibe que el movimiento del usuario en el eje Z afecta fundamentalmente a la estimación del  $PoRY$ . Además, conforme avanzamos de posición el error en Y aumenta gradualmente (ver fig. 7.6), con un valor de 18 píxeles en la posición central hasta los 67 píxeles en posiciones extremas.

Más tarde, se intenta poner en común esta información del error en Y por posiciones con el cálculo de la posición de la cabeza en el eje Z de FaceAPI. No obstante, a la vista de la imprecisión que muestran los datos de FaceAPI se empiezan a buscar otras soluciones que no dependan de esta variable para corregir la ecuación de calibración.

## 7.2 Propuesta de ecuaciones de mejora

A partir de los resultados obtenidos, y al ver que el error cometido por el desplazamiento de la cabeza en el eje Z afecta fundamentalmente a la estimación del  $PoRY$  se propone dos soluciones para corregir la ecuación de calibración: una normalización que mantenga constante la componente Y del vector medio entre la pupila y los reflejos, y la suma de un término a la ecuación del  $PoRY$  que dependa de la distancia entre los reflejos (ya que en las pruebas realizadas, esta sigue un comportamiento lineal con la posición en Z).

Ambas soluciones no requieren la información de la posición de la cabeza de FaceAPI ya que después de haber observado los resultados de las pruebas sabemos que la posición de la

cabeza en el eje Z calculada por dicho sistema es imprecisa y sería difícil mejorar la ecuación a partir de esta información. Además, la información adicional que requieren las propuestas de nuevas ecuaciones puede ser determinada por Shakti.

### 7.2.1 Normalización de $V_y$

Como ya hemos comprobado, el error que cometemos al estimar el  $PoRY$  aumenta de forma lineal al alejarnos de la posición central en la que calibramos. Para corregir esta tendencia, se va a estudiar el comportamiento del *vector medio centro de la pupila-reflejos* sin normalizar respecto a la distancia de los reflejos en las siete posiciones donde se han realizado las pruebas. La normalización del vector diferencias respecto a la distancia de los reflejos consigue que el error cometido ante desplazamientos reducidos de la cabeza con la ecuación de partida se reduzca considerablemente [1], no obstante esta normalización no consigue que este vector se mantenga constante al observar al mismo punto de la pantalla desde alejadas posiciones en el eje Z.

El propósito de este estudio es realizar una normalización del vector diferencias distinto para conseguir mantenerlo constante respecto al movimiento de la cabeza en este eje. Al cambiar la variable predictora con el desplazamiento del usuario, la ecuación de calibración produciría resultados diferentes a los de la posición central (pos. 4). Para ello tendremos que ver un comportamiento parejo de este vector en todas las posiciones de cada fijación de cada usuario. Como el movimiento de la cabeza afecta fundamentalmente a la estimación del  $PoRY$  sólo se intenta realizar la normalización del vector  $V_y$  (ver apartado 3.1).

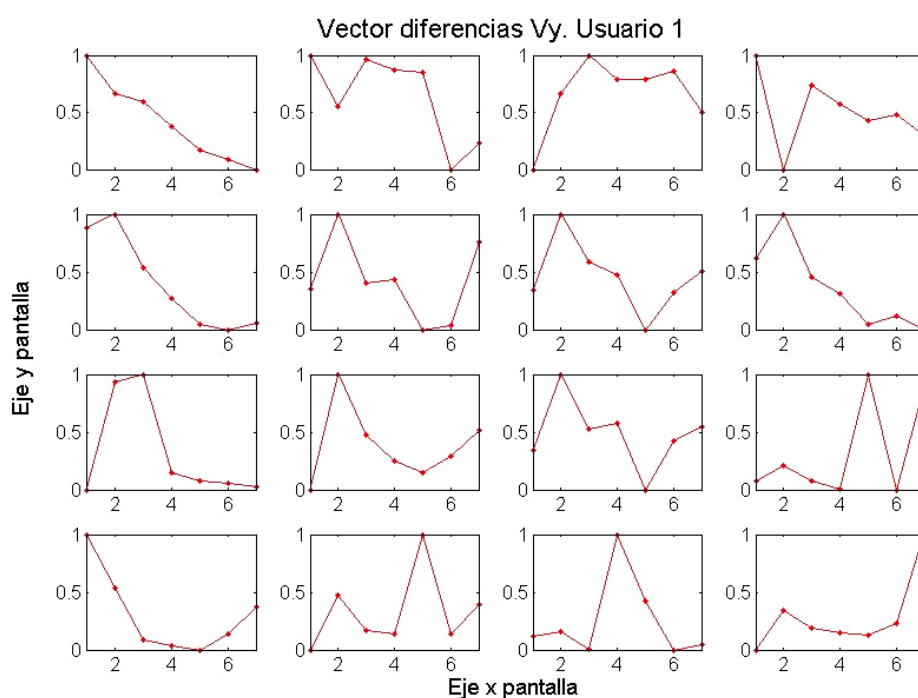
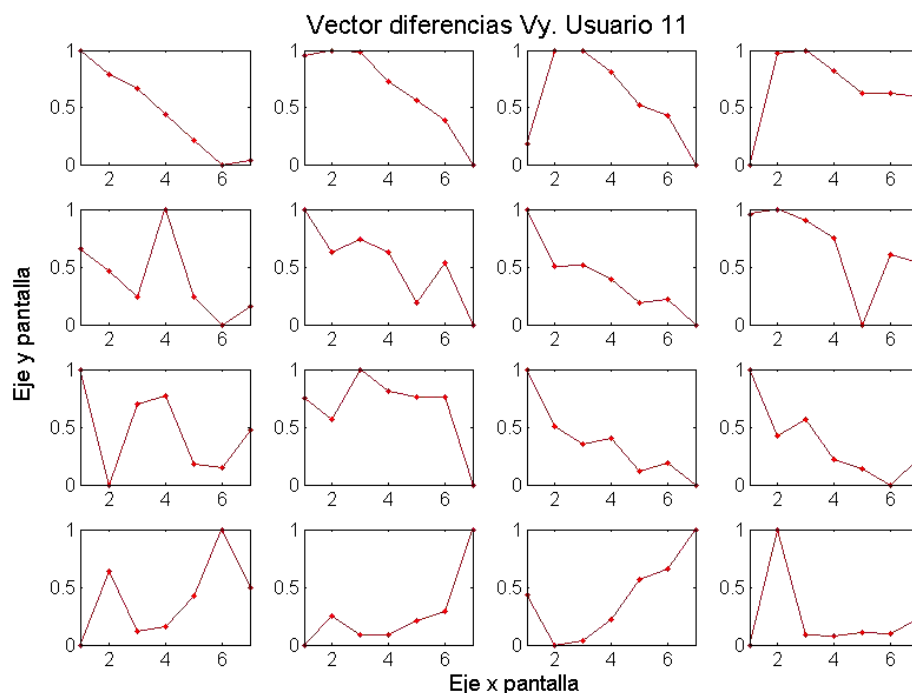
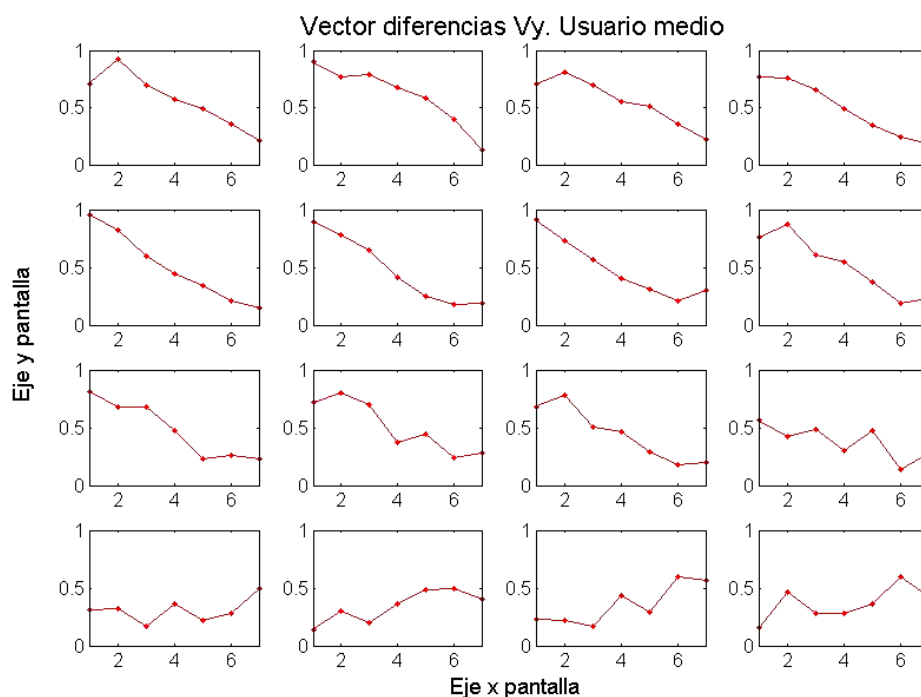


Fig. 7.17 Componente Y del vector diferencias. Usuario 1



**Fig. 7.18** *Componente Y del vector diferencias. Usuario 11*



**Fig. 7.19** *Componente Y del vector diferencias. Usuario medio*

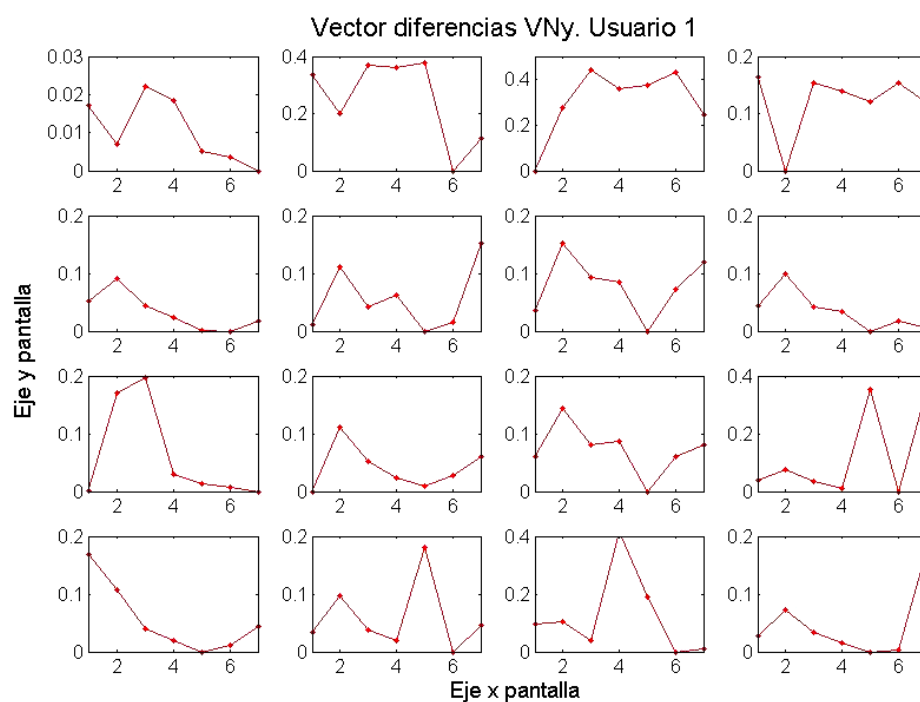
En las figuras anteriores (fig. 7.17, 7.18 y 7.19) se observa el comportamiento medio del vector  $V_y$  medio (normalizado entre 0 y 1) de cada fijación en las siete posiciones en las que se han realizado las pruebas. Cada uno de los gráficos de cada figura representa una fijación concreta (ordenadas correctamente para simular la pantalla). Cada una de estas, presenta el valor medio del vector  $V_y$  de las siete posiciones donde los usuarios han calibrado.

Las dos primeras figuras (fig. 7.17 y 7.18) corresponden a los datos obtenidos de los usuarios uno y once y la tercera (fig. 7.19) con el usuario medio.

Es difícil encontrar una tendencia clara del vector  $V_y$  para poder normalizarlo y así conseguir que se mantenga constante siempre que el usuario esté mirando al mismo punto independientemente de la posición en  $Z$  en la que se encuentre. En general, en las fijaciones de la parte superior de la pantalla, el vector  $V_y$  decrece proporcionalmente a medida que avanzamos hacia las posiciones más alejadas. En las fijaciones medias-bajas este decaimiento se vuelve menos abrupto y en las fijaciones de la parte inferior de la pantalla ocurre el efecto contrario, es decir, a medida que nos alejamos de la cámara para la calibración, el vector  $V_y$  aumenta.

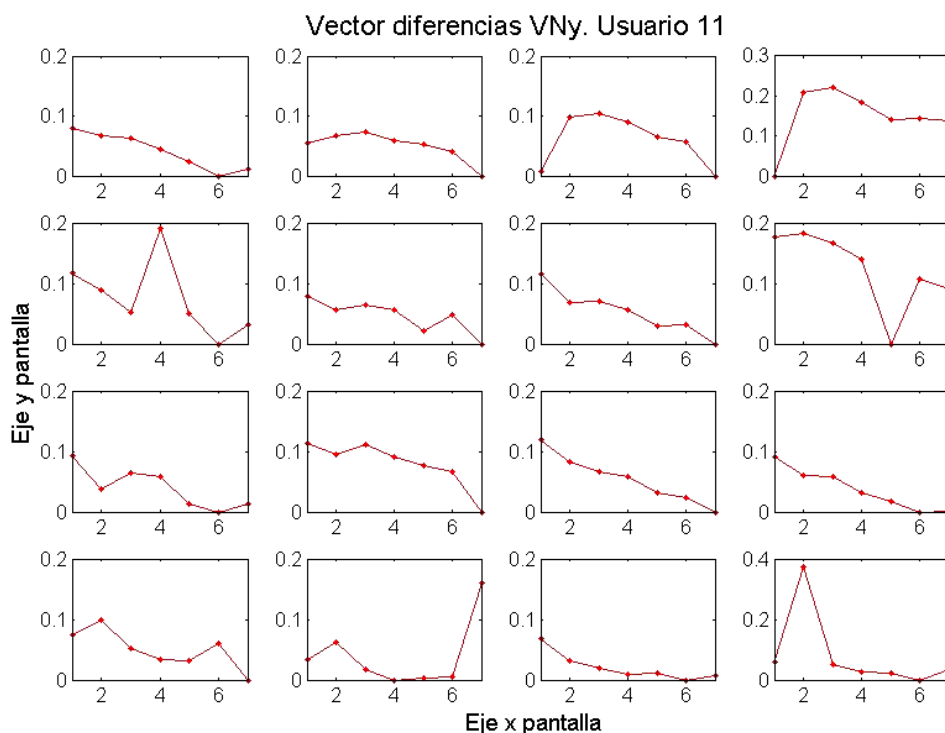
No obstante, todas las fijaciones no son equiparables a una expresión matemática común (como una recta, una parábola, etc.) incluso utilizando diferente normalización para distintos conjuntos de fijaciones.

Antes de buscar otra solución para corregir la ecuación de calibración de partida, se intenta observar el comportamiento de este vector, esta vez normalizado frente a la distancia de los reflejos y ver si a partir de ahí se puede realizar otra normalización posterior para poder corregir la ecuación de calibración. Este mismo estudio también comprueba el efecto de dicha normalización frente a desplazamientos más abruptos que los estudiados en el artículo de Cerrolaza [1].

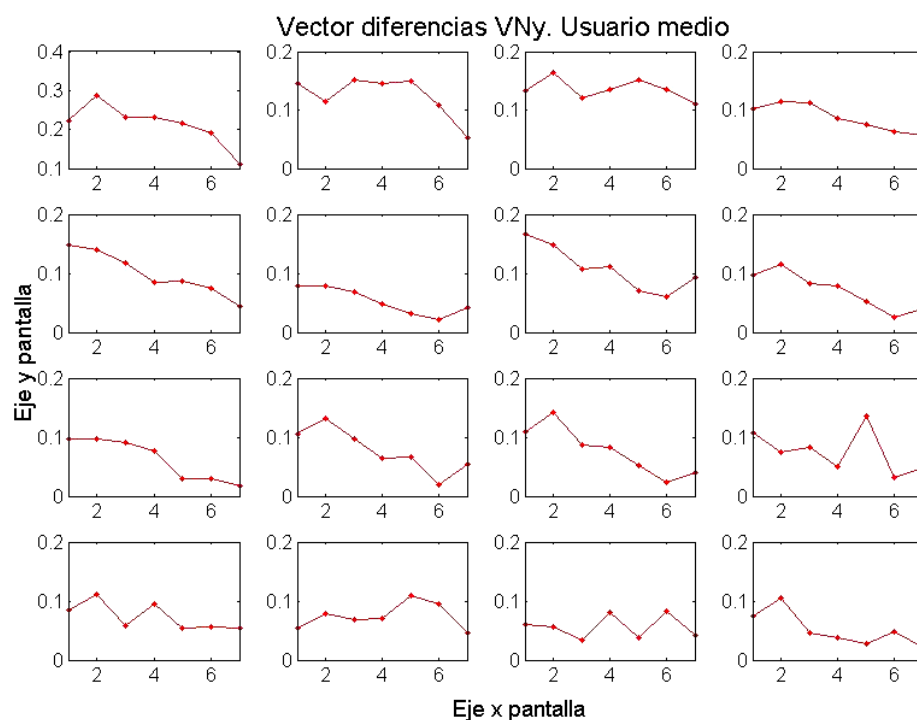


**Fig. 7.20** Componente  $Y$  del vector diferencias normalizado frente a la distancia de los reflejos.  
Usuario 1





**Fig. 7.21** *Componente Y del vector diferencias normalizado frente a la distancia de los reflejos.*  
*Usuario 11*



**Fig. 7.22** *Componente Y del vector diferencias normalizado frente a la distancia de los reflejos.*  
*Usuario medio*

En este caso tampoco se observa una tendencia clara de los datos respecto a las posiciones de calibración, de la misma manera que ocurre en las figuras 7.17, 7.18 y 7.19 por lo que desecharemos la idea de realizar una nueva normalización del vector  $V_y$  para intentar corregir la ecuación de calibración.

### 7.2.2 Añadir un término de corrección a la ecuación *PoRY*.

Otra solución factible para corregir la ecuación de calibración es intentar buscar una variable que nos de información de la posición del usuario en el eje Z. Es decir una variable que mantenga una relación lineal con la posición en Z y no varíe ante desplazamientos en el eje X o Y, y ante cambios de posición de la mirada. Una variable que cumpla estas características es la distancia entre los reflejos que podemos calcular a través de la información de Shakti. En estudios anteriores [1] ya se observó como la inclusión de este parámetro en la ecuación de regresión proporcionaba una mayor robustez al sistema ante desplazamientos en el eje Z.

Como hemos comprobado en el apartado anterior, el error en Y sigue un comportamiento lineal al desplazarnos en el eje Z, una posible estrategia de corrección sería la compensación de ese error a partir de la propia ecuación de calibración considerada (ecuación 3.1 y 3.2) a partir de la distancia entre los reflejos. Para comprobar el comportamiento lineal de esta variable con la posición en el eje Z se ha realizado un estudio de la tendencia que sigue la distancia entre los reflejos media de cada posición representando cada una de las fijaciones independientemente (fig. 7.23 y 7.24). En este caso el resultado es favorable ya que esta distancia demuestra un comportamiento muy lineal con la posición de la cabeza y además, se mantiene en todas las fijaciones.

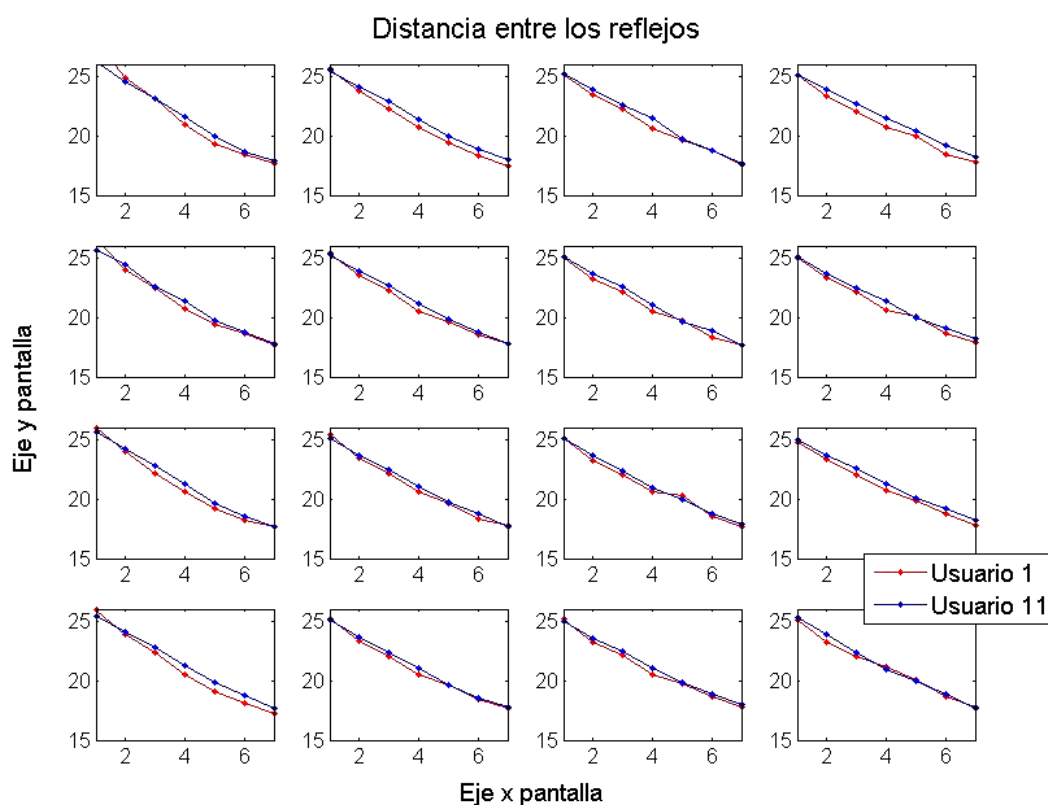
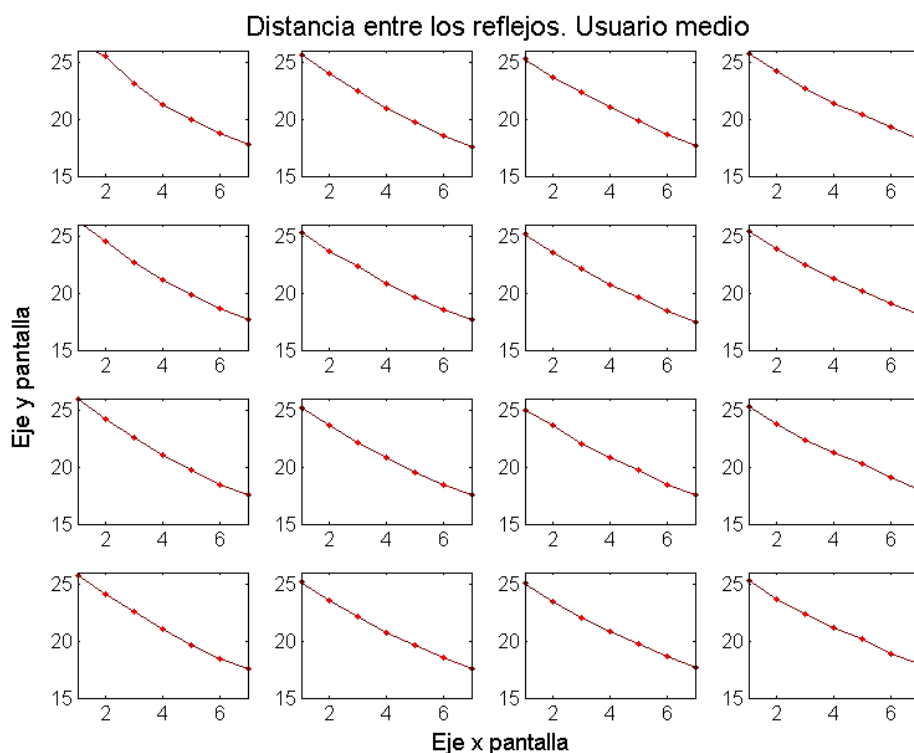


Fig. 7.23 Distancia entre los reflejos

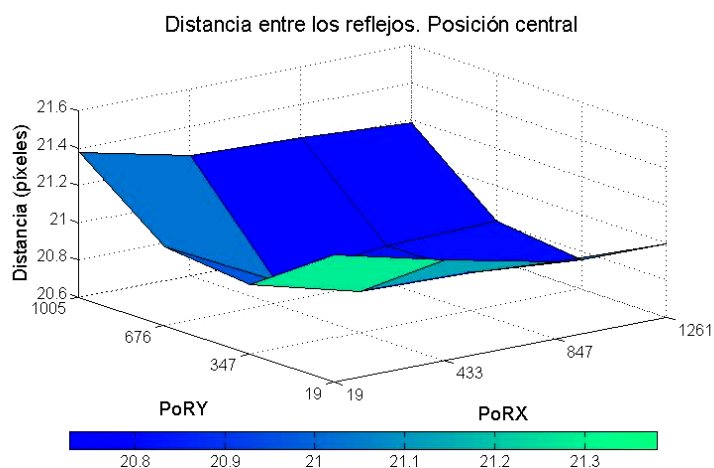
Ambas figuras (fig. 7.23 y 7.24) representan la reducción de la distancia entre los reflejos a medida que el usuario se aleja de la cámara. Este decaimiento es lineal, es decir, la disminución de la distancia de los reflejos es proporcional a la distancia del usuario a la cámara. Y al contrario de lo que sucedía con la información proporcionada por FaceAPI, el comportamiento es común para todos los usuarios.

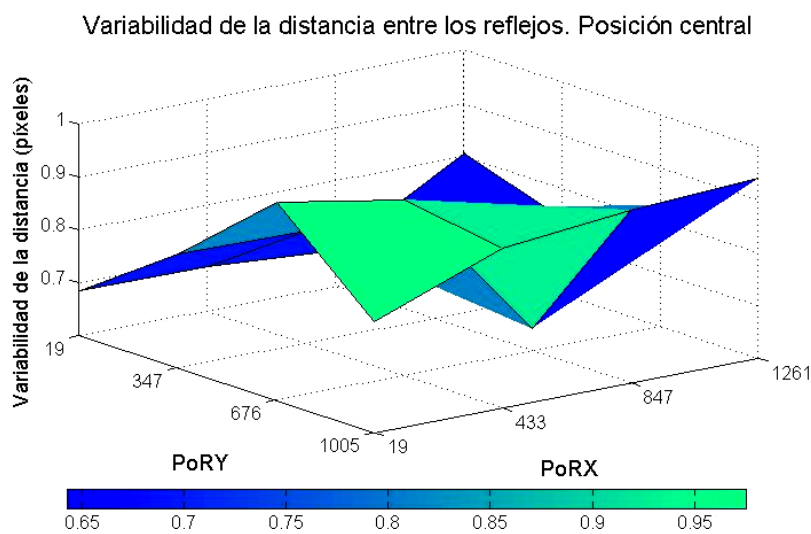
Esta variable por tanto parece un buen punto de partida para compensar el error producido por la ecuación de calibración con la distancia a la pantalla. La distancia entre los reflejos puede darnos información de la posición en el eje  $Z$  del usuario. La idea de esta solución es añadir un término adicional a la ecuación de calibración, solamente al  $PoRY$ , la componente que presenta mayor error ante los desplazamientos en  $Z$ . Este término será proporcional a la distancia de los reflejos, ya que tanto el error en el eje  $Y$  como la distancia entre los reflejos muestran un comportamiento lineal ante los desplazamientos en el eje  $Z$ .



**Fig. 7.24** Distancia entre los reflejos. Usuario medio

Además de observar el comportamiento medio en cada posición de la distancia entre los reflejos, debemos asegurarnos de que esta variable no presenta una dependencia o variación con el movimiento ocular para una posición de la cabeza fija. De no ser así, esta corrección de la ecuación de calibración no sería posible. Para ello a continuación mostramos un gráfico en donde se refleja la distancia entre los reflejos de cada fijación en la posición central (fig. 7.25), junto con la desviación de estos datos en cada fijación (fig. 7.26).



**Fig. 7.25** Distancia entre los reflejos. Usuario medio**Fig. 7.26** Variabilidad de la distancia entre los reflejos. Usuario medio

En comparación con la diferencia entre la distancia de los reflejos entre dos posiciones la variación que muestra esta variable es baja ante cambios de dirección de la mirada. La variación de la distancia media para cada fijación es de 0.4 píxeles sin embargo, la diferencia entre dos posiciones llega a ser de un píxel. Con esto, podemos validar la opción de promediar la distancia entre los reflejos para una misma posición.

### 7.2.2.1 Corrección fija de la ecuación de calibración

Al comprobar la linealidad existente entre el movimiento de la cabeza en el eje Z y la distancia de los reflejos concluimos que la distancia entre los reflejos nos da información de la posición del usuario en el eje Z y se pretende utilizar esta variable para añadir un nuevo término a la ecuación de calibración que corrija el error provocado por desplazamientos en este eje. Hay varias formas de realizar esta corrección, la primera que se va a estudiar es una corrección fija para cada usuario. Es decir, una constante fija para todos los usuarios controlada por la distancia de los reflejos más un término independiente que se calculará en cada sesión de calibración.

**Fig. 7.27** Error medio de cada posición del usuario medio en el eje Y

Para añadir el término adicional, primero se van a poner en relación la distancia entre los reflejos y el error en el eje  $Y$  de cada posición. En la gráfica anterior (fig. 7.27) mostramos el error en el eje  $Y$  medio de todas las posiciones del usuario medio, ya que anteriormente solo se ha mostrado el error absoluto en este eje.

El error en  $Y$  puede decirse que disminuye de forma gradual conforme avanzamos de posición. Por tanto el término que añadiremos será una recta (ver ecuación 7.1). Se intentará compensar el error en  $Y$ , es decir, el término añadido será una recta debido al comportamiento lineal del error ante estos desplazamientos.

$$f(D) = k * D - k_0 \quad (7.1)$$

Se trata de una función de la distancia entre los reflejos. La constante ' $k$ ' será la pendiente de la recta que se utilizará como término de corrección y el término independiente se calculará para hacer coincidir el error en la posición central, la cuarta posición, de la ecuación original con la nueva ecuación. Es decir, se quiere conseguir un error más reducido utilizando esta nueva ecuación ante desplazamientos de la cabeza, pero, manteniendo la misma ecuación para la posición central. Para ello la recta que se va a sumar a la ecuación original deberá pasar por cero cuando el usuario este en la posición donde se calibró el sistema.

Primero se calculará la media del error en  $Y$  entre la estimación del  $PoRY$  y el  $PoRY$  real de cada posición de los usuarios que han participado en las pruebas, es decir un único valor por cada posición y se promediará el valor de cada posición de todos los usuarios (fig. 7.27). El error por posición del usuario medio lo aproximaremos a una recta. Para la distancia de los reflejos procederemos de la misma manera.

A partir de estos datos podremos calcular la pendiente de la recta ( $k$ ) que se multiplicará por la distancia entre los reflejos y el término independiente ( $k_0$ ) que será calculado en cada sesión de calibración.

$$\text{Pendiente} \rightarrow k = (\text{Error}Y_7 - \text{Error}Y_1) / (D_7 - D_1) \quad (7.2)$$

$$\text{Término independiente} \rightarrow k_0 = D' * k \quad (7.3)$$

*Los subíndices indican el valor de la posición del usuario en el eje  $Z$ .*

*$D$  es la distancia entre los reflejos media por posición.  $D_1$  y  $D_7$  son la distancia media de las posiciones 1 y 7 respectivamente de los datos del usuario medio calculado en las pruebas del estudio experimental. Sin embargo,  $D'$  corresponde con la distancia entre los reflejos media de la sesión de calibración.*

El valor de  $k$  calculado a partir de las pruebas realizadas en el estudio experimental es igual a 14,9384. Y  $k_0$  será calculado expresamente en cada calibración.

La ecuación de partida se verá modificada de la siguiente manera:

$$PoRX = Cx_0 + Cx_1 V_N x + Cx_2 V_N x^3 + Cx_3 V_N y^2 \quad (7.4)$$

$$PoRY = Cy_0 + Cy_1 V_N x + Cy_2 V_N y + Cy_3 V_N x^2 V_N y + f(D) \quad (7.5)$$

$$f(D) = k * \text{Dist} - k_0 \quad (7.6)$$

Otra forma de calcular estos dos coeficientes es mediante la regresión múltiple. Para realizar la regresión las variables predictoras serán en este caso la distancia entre los reflejos media de todos los usuarios para cada posición y las variables respuesta serán el error en el eje  $Y$  medio en cada posición de todos los usuarios. (La pendiente de la recta será  $k = 15,5602$ ).

### 7.2.2.2 Corrección de la ecuación de calibración individual

Dada la elevada cantidad de usuarios y posiciones en las que se han realizado las pruebas se puede considerar la constante  $k$  del término añadido a la ecuación de calibración fijo para todos los usuarios. No obstante, el cálculo de estas variables se puede realizar de forma independiente para cada usuario. Esta solución implica la realización de una calibración diferente, la cual incluirá el movimiento en el eje  $Z$  del usuario.

Este cálculo en un principio se ha realizado de las mismas dos maneras que en el caso de la corrección fija solo que esta vez en vez de utilizar los datos de la distancia entre reflejos y error en  $Y$  del usuario medio, habrá que utilizar los datos de cada usuario. Así, el término añadido será independiente para cada usuario.

Sin embargo, esta primera aproximación al cálculo de un término individual para cada usuario resulta inviable ya que el usuario para ello debería calibrar en siete posiciones diferentes y resulta un sistema muy incómodo, por lo que la solución anterior no parece muy útil. Como estrategia alternativa, se decide calcular los coeficientes del término añadido por regresión también pero, en este caso solo se utilizará información correspondiente a la posición central y a dos de las posiciones más cercanas a esta.

Aprovechando que el número de posiciones en las que se han realizado las pruebas es alto, de la misma manera que en el caso anterior se han calculado los coeficientes pero esta vez con la información del error en el eje  $Y$  y la distancia entre los reflejos, de las posiciones 2, 4 y 6.

Por último para tratar de optimizar el coste temporal del proceso de calibración, se ha realizado otro cálculo de estos coeficientes con la información de las posiciones 2, 4 y 6 pero, en el caso de las posiciones 2 y 6 solo con la información de las cuatro fijaciones centrales de la pantalla.

### 7.2.3 Conclusiones

En vista de que el error cometido al estimar el  $PoRY$  con el desplazamiento en el eje  $Z$  aumenta de forma gradual se han propuesto dos soluciones para corregir la ecuación de calibración y permitir al usuario movimiento libre en este eje.

La primera de ellas se trata de normalizar el vector pupila reflejos  $V_y$  para mantenerlo constante ante estos desplazamientos. Al resultar difícil esta solución se intenta buscar una variable que mantenga una relación lineal con la posición en  $Z$  y no varíe ante desplazamientos en otros ejes ni al cambiar la dirección de la mirada: la distancia entre los reflejos.

Esta variable por tanto será el punto de partida para compensar el error producido por la ecuación de calibración con la distancia a la pantalla añadiendo a la ecuación de calibración  $PoRY$  un término proporcional a esta. La nueva ecuación corresponde con una recta de compensación (ya que el error en  $Y$  es proporcional a la distancia en el eje  $Z$ ), es decir, una constante que multiplica con la distancia de los reflejos más un término independiente.

Principalmente hemos escogido dos formas de calcular estos términos: mediante la información del usuario medio con el que se realizaron las pruebas y de forma independiente para cada usuario.

## 7.3 Aplicación de las nuevas ecuaciones

En el apartado anterior (7.2) se proponen dos modos de corrección de la ecuación de partida: mediante la información del error en  $Y$  y la distancia entre los reflejos por posiciones del usuario medio de las pruebas realizadas, y otra calculando la corrección individualmente en

cada sesión. En este apartado pasaremos a una nueva estimación del  $PoR$  a través de estas ecuaciones utilizando los datos de las pruebas del estudio experimental.

### 7.3.1 Error medio de cada posición

Primero, vamos a calcular el error medio de la estimación del  $PoRY$  respecto al  $PoRY$  real de cada posición. Así podremos comprobar el efecto de cada una de las ecuaciones sobre los datos correspondientes a las pruebas del estudio experimental.

Para esta elección se tomará como referencia el error medio y máximo de cada posición y se promediarán todos los usuarios obteniendo el error del usuario medio en las siete posiciones. También se calculará la variabilidad del error en cada posición como la diferencia entre el error medio de cada una de las fijaciones máximo y mínimo.

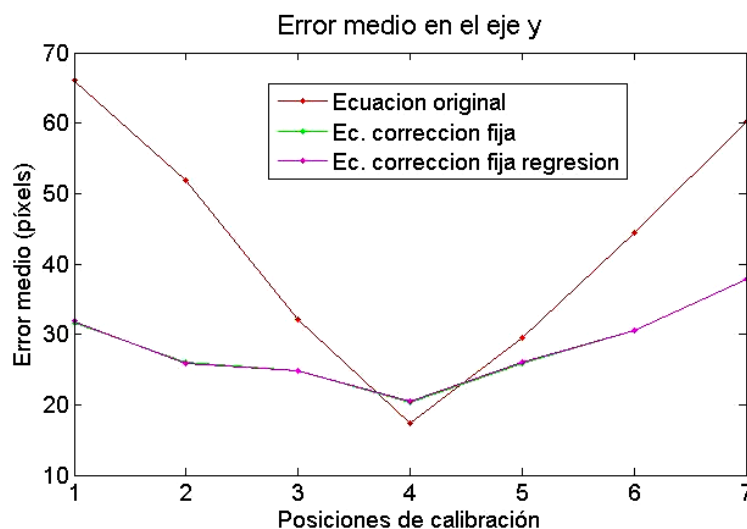


Fig. 7.28 Error medio. Ecuaciones con corrección fija

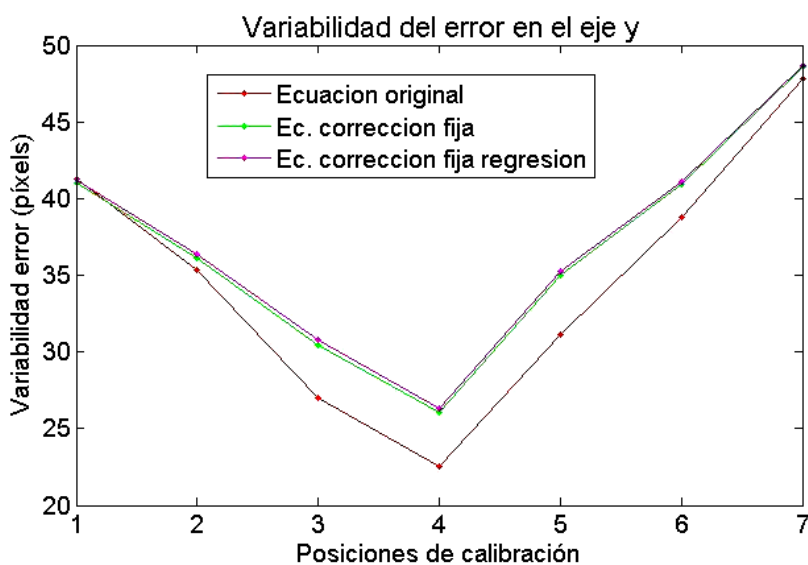


Fig. 7.29 Variabilidad del error medio. Ecuaciones con corrección fija

En estas dos primeras figuras (fig. 47 y 48), se compara la ecuación de partida, la original, con las nuevas ecuaciones corregidas respecto a la distancia de los reflejos de cada posición del usuario. En la primera (fig. 47) se observa el comportamiento de las ecuaciones con corrección fija, las cuales se comportan muy parecido según se observa en la gráfica. Eso sí, ambas obtienen un error más bajo que la ecuación original a medida que nos alejamos de la posición central, siendo más evidente en las posiciones extremas. Sin embargo, el error de la ecuación original en la posición central está 3 píxeles por debajo de de las ecuaciones corregidas. No obstante, la cuantía de la diferencia entre los errores de cada ecuación dependerá de la variabilidad del error que presenta cada ecuación en esta posición. Viendo que esta variabilidad está por encima de los 20 píxeles para las tres ecuaciones, comprobamos que este cambio en la posición central no es significativo.

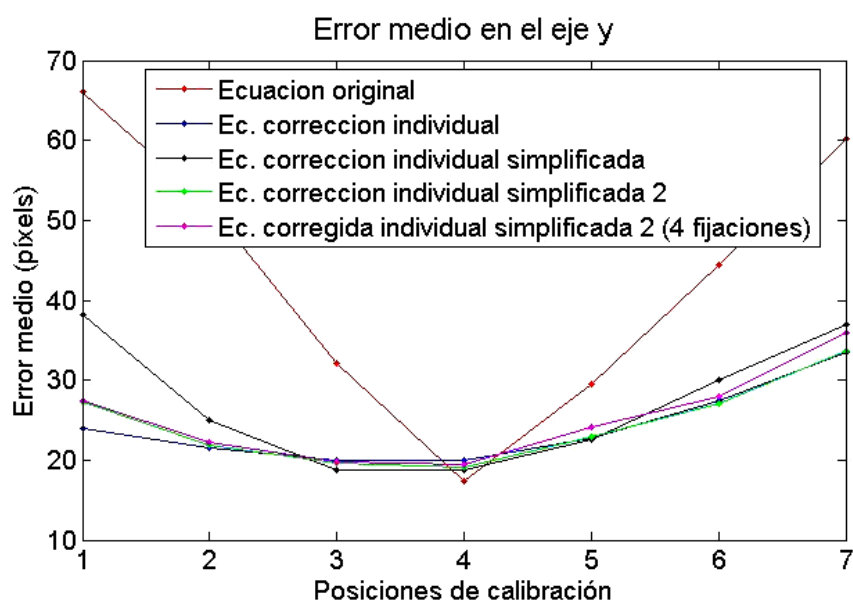


Fig. 7.30 Error medio. Ecuaciones con corrección individual

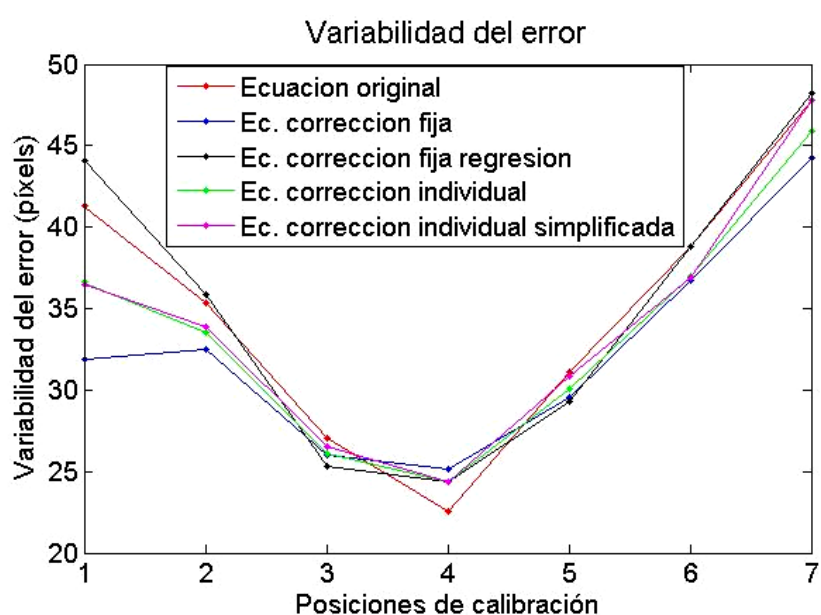
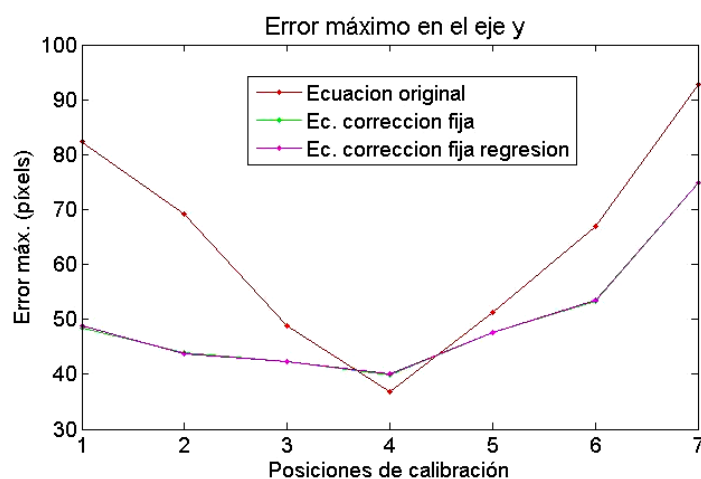


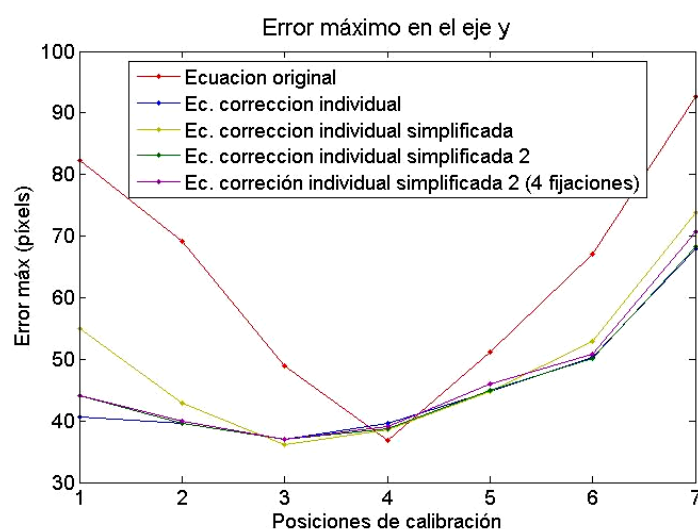
Fig. 7.31 Variabilidad del error medio. Ecuaciones con corrección individual



En las dos figuras anteriores (fig. 49 y 50) que muestran la comparación entre las ecuaciones de corrección individual y la original se obtienen las mismas conclusiones que para las de corrección fija. Es decir, el error de las nuevas ecuaciones es menor al de la original aumentando conforme nos alejamos de la posición central. Lo interesante en la figura 49 es la diferencia entre cada una de las correcciones. La corrección individual por regresión es la que se tienen en cuenta todas las posiciones consigue el error más bajo sin embargo, esta ecuación no es apropiada ya que requiere de un tiempo de calibración muy elevado, además sería complicado conseguir el seguimiento en un recorrido de 12cm en el eje Z. Habiendo desechado esta opción, se estudian las correcciones simplificadas: el término adicional añadido a la ecuación de calibración de la primera simplificada está calculado a partir de la información de las posiciones 3, 4 y 5 y la segunda a partir de la información de las posiciones 2, 4 y 6. La primera mantiene el error más bajo que la segunda en las posiciones centrales, sin embargo en las posiciones extremas no. La última ecuación, la cual está construida a partir de la información de la posición 4 y de las 4 fijaciones centrales de las posiciones 2 y 6, ofrece un resultado muy similar al obtenido con la ecuación anterior. A continuación se muestra el error máximo cometido al estimar el *PoR* con las diferentes ecuaciones:



**Fig. 7.32** Error máximo. Ecuaciones con corrección fija



**Fig. 7.33** Error máximo. Ecuaciones con corrección individual

El error máximo de todas las ecuaciones comprobadas (fig. 51 y 52) sigue el mismo comportamiento que el error medio para todas las ecuaciones, siendo este valor más elevado.

Entre las ecuaciones de corrección fija e individual obtenemos unos resultados muy parecidos del error cometido en la estimación del *PoRY*. El mejor de los resultados lo ofrece la ecuación con corrección individual por regresión en la que se tienen en cuenta todas las posiciones pero esta no es una solución viable por el tiempo de calibración que requiere. Y entre las otras tres soluciones de corrección individual en las que solo se toman en cuenta tres de las posiciones, al obtener resultados tan similares, la más adecuada será la última, la que menos tiempo de calibración precisa. Sin embargo, a pesar de que con la corrección individual parece que se obtienen mejores resultados, fijándonos en la variabilidad del error que para todos los casos supera los 20 píxeles, vemos que la diferencia no es significativa. Además la ecuación con corrección fija presenta una ventaja frente a la de corrección individual ya que no hará falta aumentar el tiempo por la necesidad de la calibración en dos puntos adicionales a la posición donde se calibra la cámara.

No obstante, dependiendo del fin para el que el usuario vaya a utilizar el sistema de eye-tracking será más apropiado el uso de una ecuación o de otra. Es decir, por ejemplo para aplicaciones en las que el usuario requiera un recorrido reducido de movimientos de cabeza bastará con la ecuación de corrección fija o incluso con corrección individual, pero realizando la calibración de los dos extremos en distancias próximas a la central (unos 2cm). Sin embargo, para que el sistema permita mayores desplazamientos de la cabeza, será necesaria la calibración individual en la cual los puntos extremos estén más separados.

### 7.3.2 Estudio estadístico de las ecuaciones corregidas

Además del estudio gráfico del comportamiento de las diferentes correcciones propuestas se va a realizar un estudio estadístico que permita comprobar la diferencia existente entre dichas ecuaciones (Paired sample ttest).

Este estudio consiste en realizar un test de la hipótesis nula de que los datos de la diferencia entre dos de las ecuaciones son una muestra aleatoria de una distribución normal con media 0 y varianza desconocida, frente a la alternativa de que la media no es 0. Si el resultado de la prueba es  $H = 1$  indica un rechazo de la hipótesis nula al nivel de confianza del 5% y si  $H = 0$  indica un no se rechaza la hipótesis nula con un nivel de confianza del 5%.

H	c	Ecuaciones a comparar
1	(13.0 , 16.7)	Ecuación original/ Ecuación de corrección fija
1	(3.1 , 4.9)	Ec. corrección fija/ Ec. corrección individual
0		Ec. corrección fija/ Ec. corrección fija por regresión
1	(-3.9 , -2.2)	Ec. corrección fija / Ec. corrección fija simplificada 1 (posiciones 3,4,5)
0		Ec. corrección fija / Ec. corrección fija simplificada 2 (posiciones 2,4,6)
1	(-1.2,-0.4)	Ec. corrección fija simplificada 2/La misma considerando 4 fijaciones (posiciones 2,6)
1	(2.0,3.8)	Ec. corrección fija/ Ec. corrección individual simplificada 2 considerando 4 fijaciones (posiciones 2,6)
1	(-2.5, -1.2)	Ec. corrección individual / Ec. corrección fija simplificada 1 *

**Tabla VII.** Resultados del test de paridad.

H = Resultado del test

$c$  = Intervalo de diferencia de poblaciones media. Esta variable indica entre las ecuaciones que se consideran diferentes por el test cual es la que mayor error presenta. Si el intervalo es negativo la segunda ecuación será la que mayor error cometa y si es positivo la primera.

Para realizar el test se han utilizado los datos del error cometido al estimar el *PoRY* de todas las fijaciones de todas las posiciones de todos los usuarios que realizaron las pruebas. Excepto en la última comparación en la cual se ha tenido en cuenta únicamente el error de las posiciones 3, 4 y 5(\*).

La primera aplicación del estudio ha sido para la ecuación original y la de corrección fija en el que el test rechaza la hipótesis nula, es decir el error cometido por cada ecuación es diferente. Además sabemos que el error de la primera ecuación es mucho mayor. Más tarde se han comparado las ecuaciones de corrección fija con la de corrección individual, y el resultado indica que la primera presenta mayor error al estimar *PoRY*.

Por otro lado se ha realizado el test comparando la ecuación con corrección individual calculada a partir de la información de las siete posiciones donde los usuarios calibraron al hacer las pruebas, con las ecuaciones que solo consideraban tres de estas posiciones.

La primera comparación con la ecuación calculada a partir de los datos de las posiciones 3, 4 y 5 rechaza la hipótesis nula, es decir, no se considera que los resultados obtenidos de ambas ecuaciones obtengan el mismo resultado. Además, el test demuestra que la media del error entre la estimación del *PoRY* y el real de la ecuación simplificada es mayor por tanto esta ecuación se considera peor, como era de esperar. En cambio, si para realizar el test solo tenemos en cuenta el error cometido en las tres posiciones centrales el resultado cambia. El test sigue rechazando la hipótesis nula, sin embargo esta vez considera que la primera ecuación provoca un error mayor (\*).

El test realizado con la ecuación que solo considera para el cálculo del término añadido a la nueva ecuación las posiciones 2, 4 y 6 no rechaza la hipótesis nula, es decir, ambas ecuaciones obtendrán los mismos resultados. No obstante, al realizar el test con la ecuación en la que el término añadido se calcula a partir de las posiciones anteriores pero considerando solo cuatro de las fijaciones en las posiciones extremas, el test rechaza la hipótesis nula, además sabemos que el error obtenido con esta ecuación es superior.

Tomando como punto de partida para escoger alguna de las ecuaciones el tiempo requerido en la calibración se va a realizar un estudio de las ecuaciones corregidas a partir de una corrección fija y la ecuación con corrección individual que implica el menor tiempo posible para la calibración, ya que aunque el resultado sea un poco peor al obtenido con otras ecuaciones la optimización del tiempo de calibración es uno de los aspectos más buscados en estos sistemas.

Por tanto, se realiza un nuevo test entre estas dos ecuaciones escogidas en el que la hipótesis nula es rechazada y además la ecuación con corrección fija presenta mayor error. A pesar de que sean diferentes en la realización de este test el error medio de cada posición obtenido por las dos ecuaciones es similar ya que la variabilidad del error es muy alta. Por tanto, ambas soluciones podrían considerarse válidas. Una presenta mayor precisión (ecuación con corrección individual) pero, supone un aumento del tiempo de calibración.

### 7.3.3 Conclusiones

En este apartado se ha realizado un estudio de todas las posibles correcciones aplicables a la ecuación de partida.

En general distinguimos dos tipos de corrección: una fija para todos los usuarios y otra corrección en la que este término será calculado en la sesión de calibración. No obstante a pesar de existir muchas variantes para calcular los términos de corrección específica para cada

usuario sólo se considera válida la que considera la información de calibración en una posición central y de cuatro fijaciones de dos posiciones más (adelante y atrás de la central).

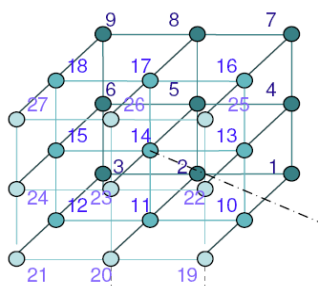
Ambas soluciones mejoran notablemente la precisión de la ecuación original, la cual en las posiciones extremas de las pruebas experimentales presenta un error de 68píxeles equivalente a una precisión aproximada de  $1.85 \pm 0.55^\circ$ . La de corrección fija es menos precisa que la de corrección individual (el error en  $Y$  es de 36píxeles y 32píxeles respectivamente en las posiciones extremas, lo cual equivale a una precisión aproximada de  $0.99 \pm 0.55^\circ$  y  $0.89 \pm 0.55^\circ$ ) no obstante, la utilización de la segunda ecuación implica aumentar el tiempo de calibración. Por tanto, dependiendo del objetivo para el que se quiera utilizar el sistema será aconsejable una u otra.

En el caso de que el tiempo de calibración sea la mayor prioridad será mejor la ecuación con corrección fija. De lo contrario si se quiere más precisión para realizar mayores movimientos será mejor la segunda opción.

## 7.4 Estudio de las nuevas ecuaciones ante desplazamientos en el eje $X$ e $Y$

En este proyecto solo se han estudiado los desplazamientos de un usuario en el eje  $Z$  y se ha propuesto una ecuación de calibración que corrija el error cometido en la estimación del  $PoRY$  cuando el usuario presente movimientos en este eje. Una vez comprobado el funcionamiento de las correcciones de la ecuación propuestas ante estos movimientos nos falta comprobar que esta modificación no empeora la estimación del  $PoR$  ante desplazamientos de la cabeza en el eje  $X$  o  $Y$ .

Para comprobarlo se han realizado nuevas pruebas las que combinen desplazamientos de usuarios en los tres ejes. Se considerarán 27 posiciones para realizar una calibración en un cubo de 6cm de arista. Entre dos posiciones consecutivas por tanto habrá una distancia de 3cm.



En la figura anterior (fig. 7.34) se exponen las posiciones consideradas y numeradas en las que se realizará una calibración por cada usuario. El sistema estará colocado en frente del cubo del lado formado por las esquinas 1, 7, 19, 25.

Fig. 7.34 Cubo posiciones de calibración

### 7.4.1 Datos de usuarios reales

Para realizar estas pruebas con usuarios reales se ha utilizado el mismo método que seguimos para las pruebas anteriores en cuanto a mantener fijas las posiciones de la cámara, las del chinrest y la de la pantalla para todos los usuarios permitiendo mover para cada usuario los iluminadores, y también se van a realizar calibraciones de 16 puntos en cada posición. Las pruebas se han realizado para tres usuarios distintos por la complejidad que conllevan.

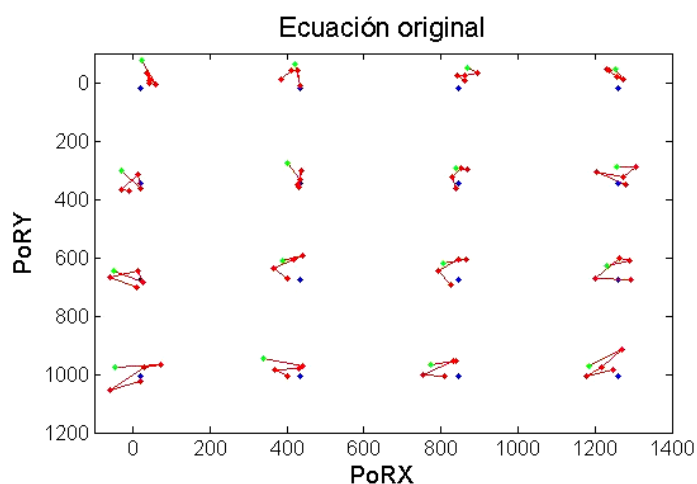
Por la experiencia de las pruebas realizadas anteriormente en las que únicamente se consideran 7 posiciones presuponemos que la duración de estas pruebas será elevada. Por tanto, se decide realizar un filtrado de las posiciones en las que habrá que realizar una calibración, concretamente por cada posición en el eje  $Z$ , solo se considerarán cinco posiciones de las nueve posibles: la posición central y las cuatro esquinas. Se decide estudiar únicamente las cuatro esquinas además de la posición central ya que contienen el error producido por la

combinación de un desplazamiento en el eje  $Y$  y otro en el eje  $X$  respecto a la posición central. Esta restricción no nos permite contemplar la contribución del error de los desplazamientos en cada uno de los ejes  $X$  e  $Y$  por separado. Aunque no nos permita estudiar movimientos independientes de los ejes, sí que nos sirve para evaluar y estudiar el comportamiento de las nuevas ecuaciones frente a otros desplazamientos.

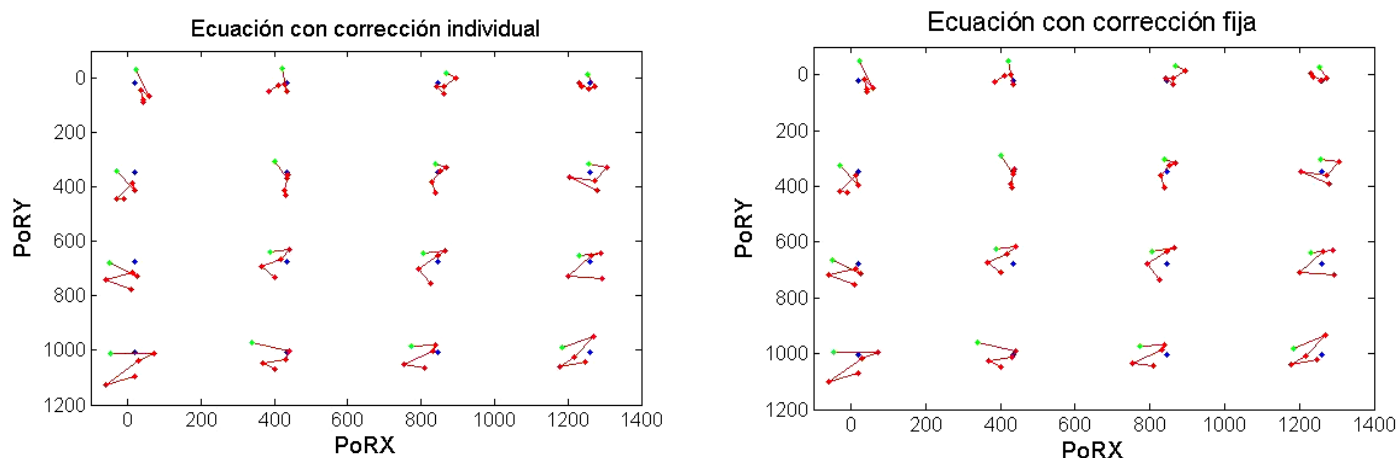
Para comprobar el resultado de las pruebas realizadas, se va a representar la estimación del  $PoR$  a través de la ecuación original por un lado, por la ecuación propuesta de corrección fija y por la de corrección individual (cada una de las tres en un gráfico distinto) frente al  $PoR$  real.

Para facilitar la visualización de los resultados se va a representar la media de la estimación del  $PoR$  de cada fijación. Pero, al ser un número elevado de posiciones se van a separar en tres todas las posiciones en las que se han realizado calibraciones, una figura por cada posición en  $Z$ . Por tanto se dibujarán 5 posiciones en cada figura ordenadas las tres del mismo modo. En la primera posición del eje  $Z$  (la más cercana a la cámara) el orden en el que serán representadas todas las fijaciones será: 7, 25, 13, 1 y 19 (numeración correspondiente a la figura 7.34).

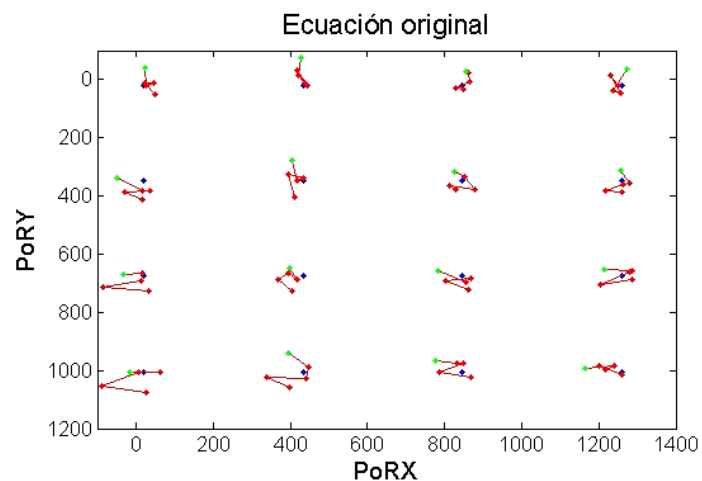
Para este primer ejemplo (fig.7.35) se ha mostrado el comportamiento de la estimación del  $PoR$  a partir de los datos del usuario 3.



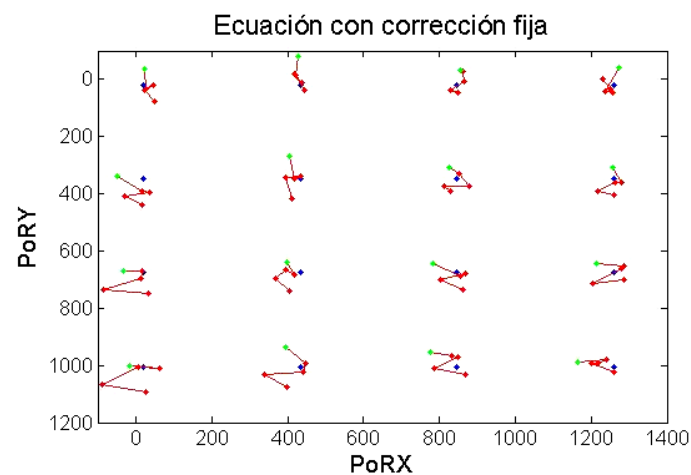
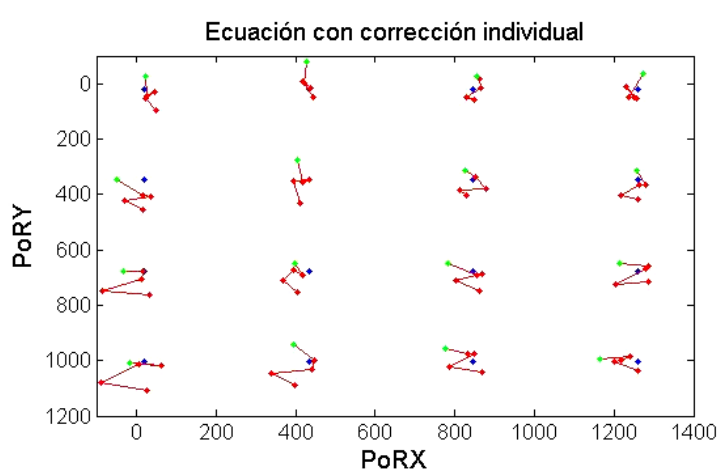
**Fig. 7.35a** Estimación del  $PoR$ . Usuario 3 en la posición delantera. Ecuación original



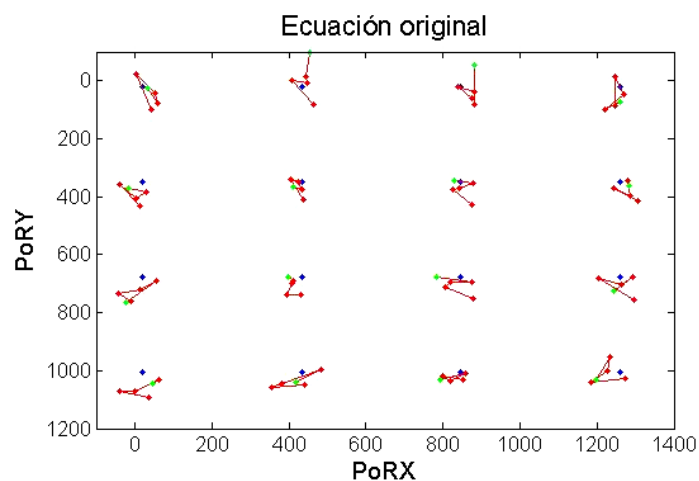
**Fig. 7.35b** Estimación del  $PoR$ . Usuario 3 en la posición delantera. Ecuaciones propuestas



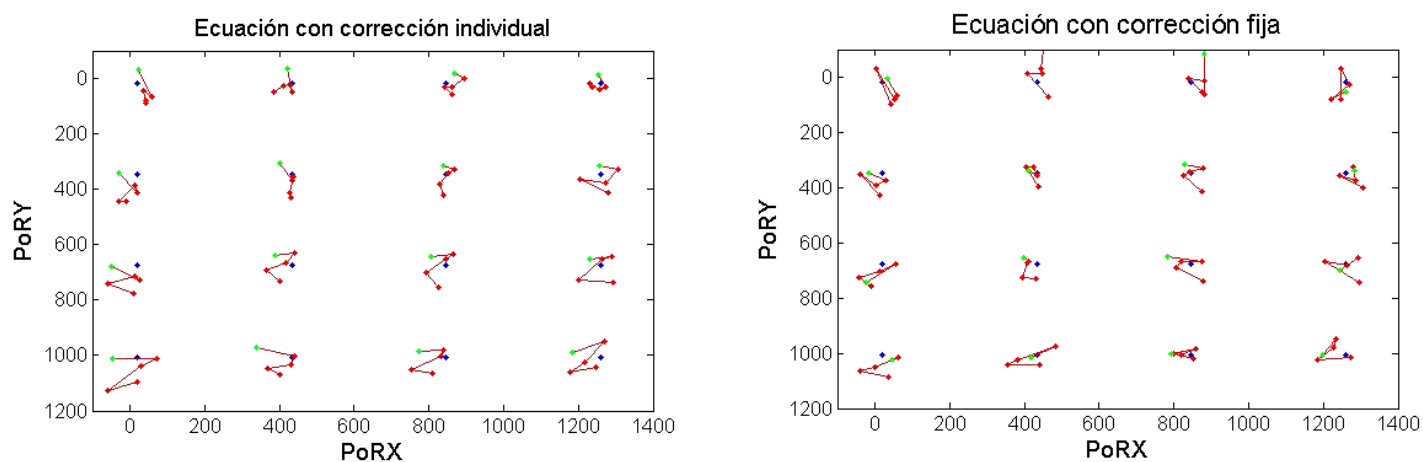
**Fig. 7.36a** Estimación del PoR. Usuario 3 en la posición central. Ecuación original



**Fig. 7.36b** Estimación del PoR. Usuario 3 en la posición central. Ecuaciones propuestas



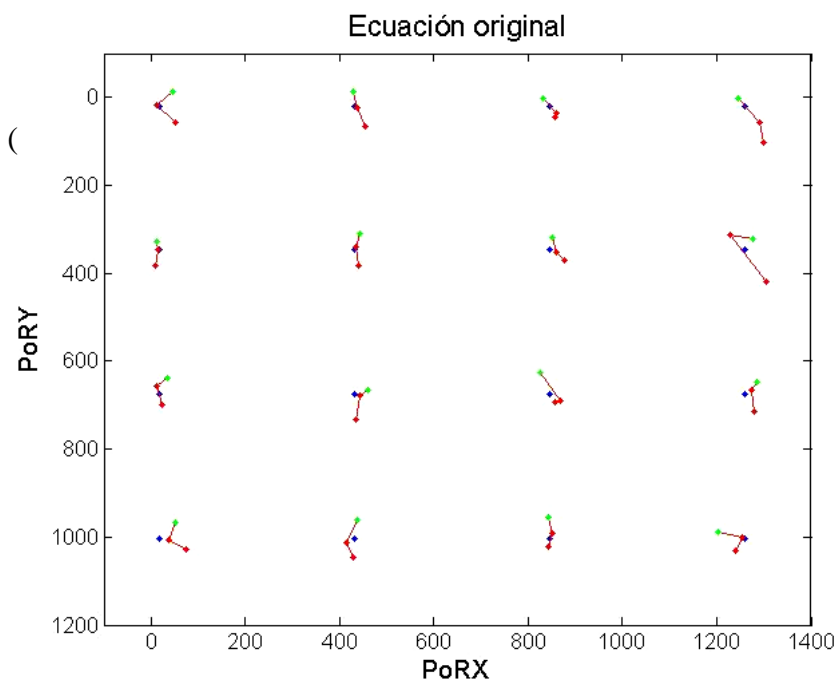
**Fig. 7.37a** Estimación del PoR. Usuario 3 en la posición trasera. Ecuación original



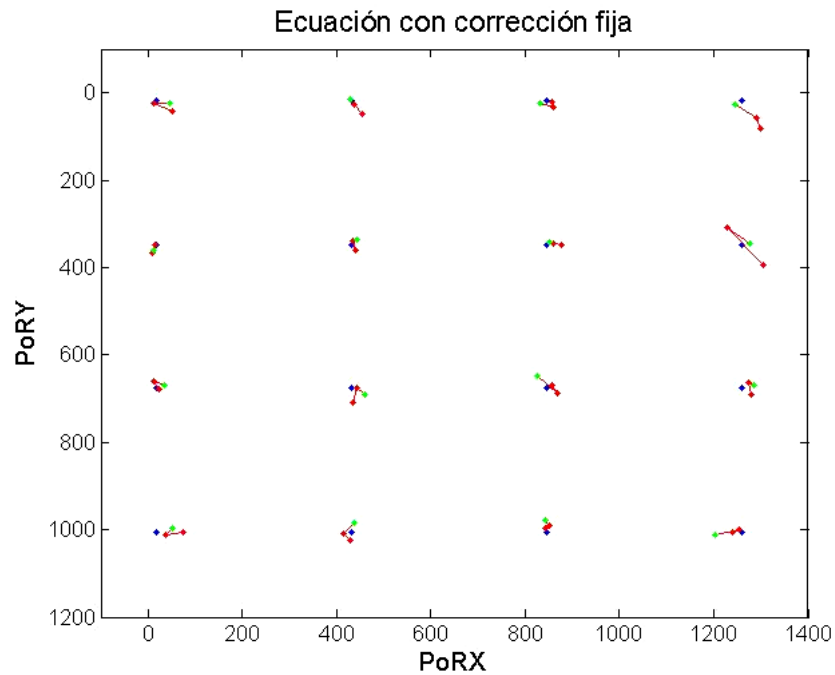
**Fig. 7.37b** Estimación del  $PoR$ . Usuario 3 en la posición trasera. Ecuaciones propuestas

En general, observando la estimación del  $PoR$  de las tres ecuaciones observamos que la variación producida por la corrección de las nuevas ecuaciones propuestas es mínima (ver fig. 7.35, 7.36 y 7.37), además, en la posición central, 14, las tres ecuaciones se comportan prácticamente de la misma manera, ya que las ecuaciones corregidas están previstas para ello.

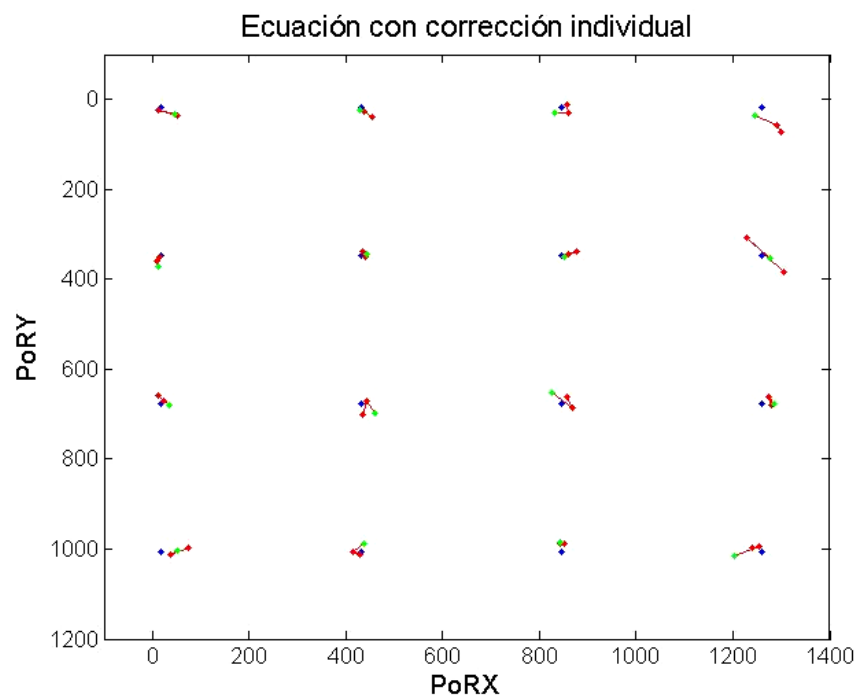
Por último comprobaremos que en las posiciones centrales, es decir con la variación en  $Z$ , el efecto de las ecuaciones propuestas es beneficioso. En la siguientes tres gráficas se observa el comportamiento de uno de los usuarios en las posiciones 13, 14 y 15, las posiciones centrales. El comportamiento seguido en el caso de los tres usuarios es el mismo que se muestra en este caso, el cual coincide con las pruebas anteriores. Por lo que la mejoría del resultado de las nuevas ecuaciones frente a este movimiento es notable (ver fig.7.38).



**Fig. 7.38 a** Estimación del  $PoR$ . Usuario 1 desplazamiento en  $Z$ .



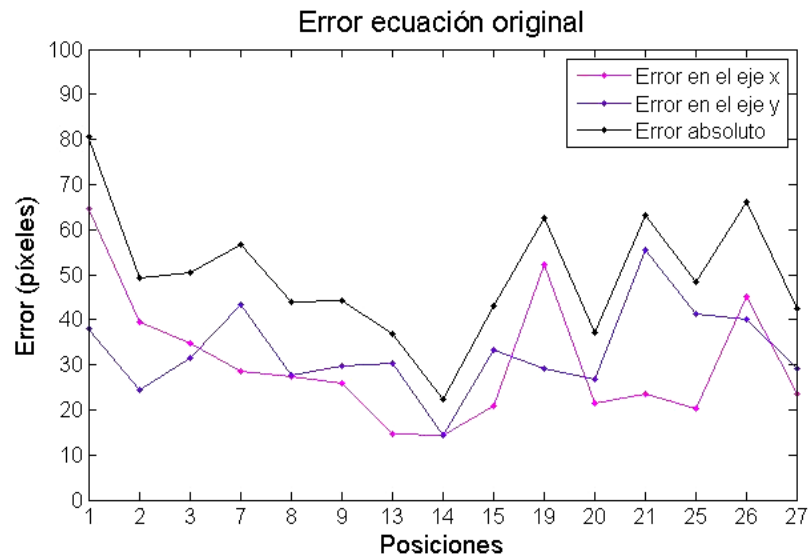
**Fig. 7.38b** Estimación del PoR. Usuario 1 desplazamiento en Z.



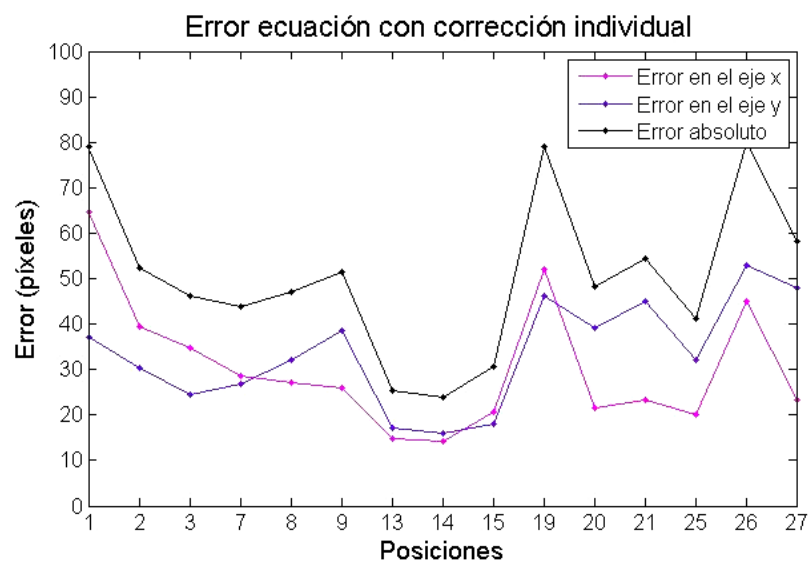
**Fig.7.38c** Estimación del PoR. Usuario 1 desplazamiento en Z.

Tras observar el comportamiento de las nuevas ecuaciones frente a combinaciones de desplazamientos en todos los sentidos, ahora vamos a cuantificar el error cometido por dichas ecuaciones gráficamente observando el error medio de toda la pantalla de cada posición.

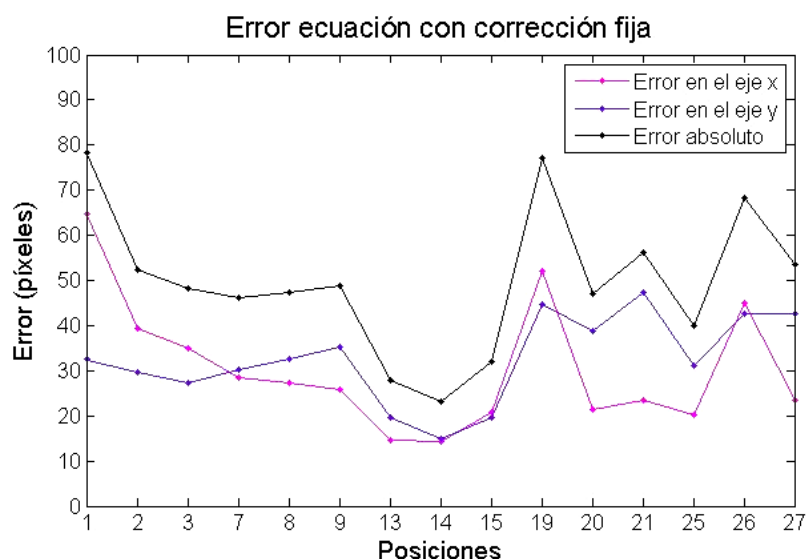




**Fig. 7.39** Error medio por posición. Ecuación original



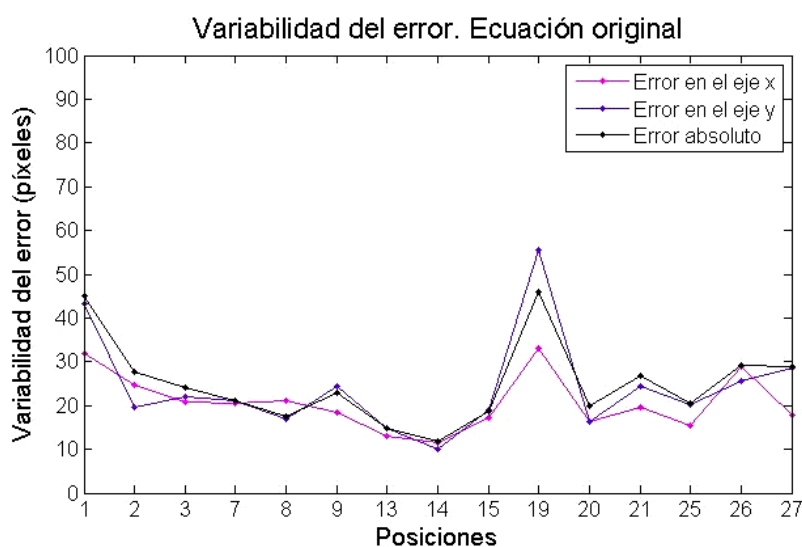
**Fig. 7.40** Error medio por posición. Ecuación con corrección individual



**Fig. 7.41** Error medio por posición. Ecuación con corrección fija

Obviamente, el error en el eje  $X$  es el mismo para las tres ecuaciones ya que la estimación del  $PoRX$  es la misma; en las ecuaciones propuestas la corrección se produce en la ecuación del  $PoRY$ . El error en el eje  $Y$  por tanto es diferente para cada una de las ecuaciones no obstante para poder cuantificarlo habrá que valorarlo teniendo en cuenta la desviación del error de cada posición. (Ver fig. 7.39, 7.40 y 7.41).

A continuación se muestra la variabilidad del error medio de todas las fijaciones en cada posición.



**Fig. 7.42** Variabilidad del error. Ecuación original

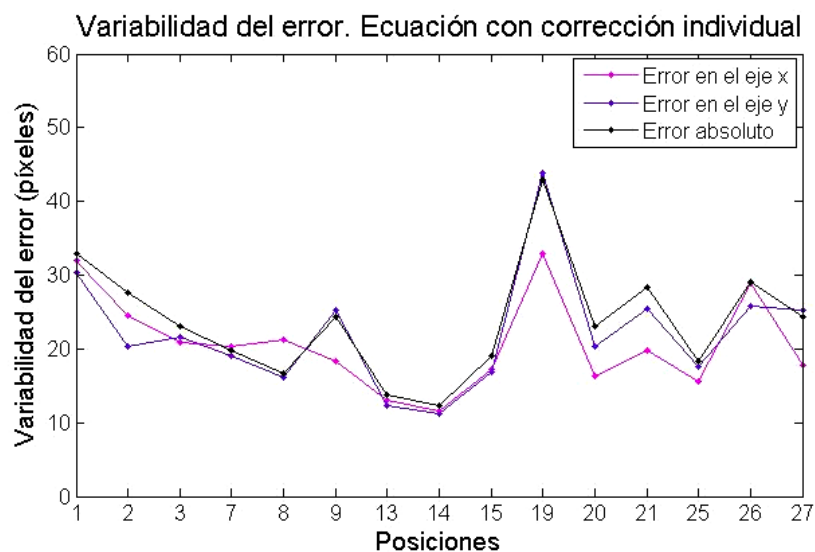


Fig. 7.43 Variabilidad del error. Ecuación con corrección fija

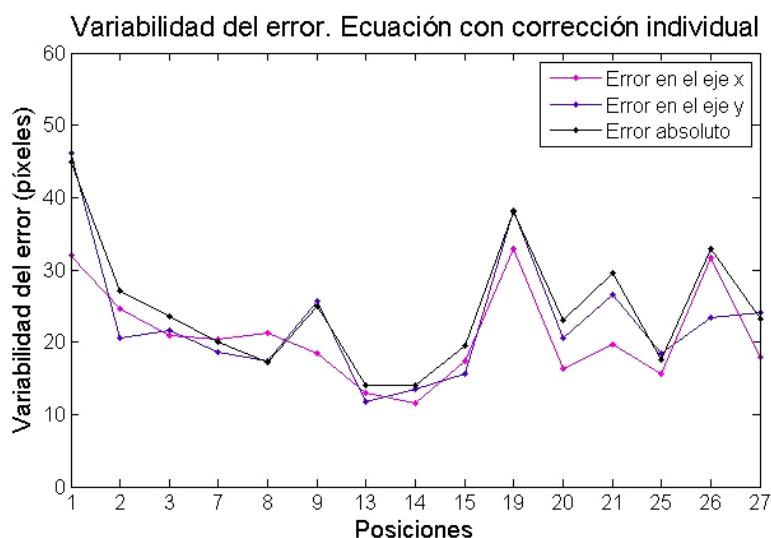


Fig. 7.44 Variabilidad del error. Ecuación con corrección individual

En general con las nuevas ecuaciones propuestas el error en el eje  $Y$  disminuye al producirse desplazamientos en el eje  $Z$ , sin embargo, ante la combinación de diferentes desplazamientos no se perciben cambios significativos en el error al atender a la variabilidad del error de cada posición.

Pero, siguiendo el objetivo de este estudio concluimos que la corrección propuesta en las nuevas ecuaciones no perjudica la estimación del  $PoR$  al producirse desplazamientos del usuario en los ejes  $X$  e  $Y$ . (Ver fig. 7.42, 7.43 y 7.44).

Otra forma alternativa de comprobar el efecto que puedan causar las ecuaciones propuestas es calcular la distancia entre los reflejos media de cada posición ya que el término añadido de ambas funciones propuestas depende directamente de este parámetro.

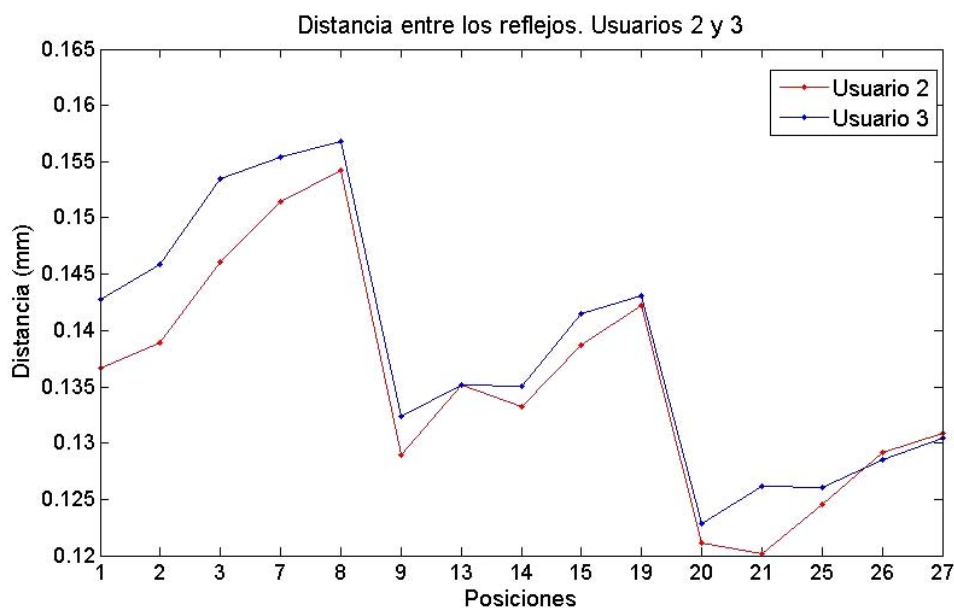


Fig. 7.45 Distancia entre los reflejos

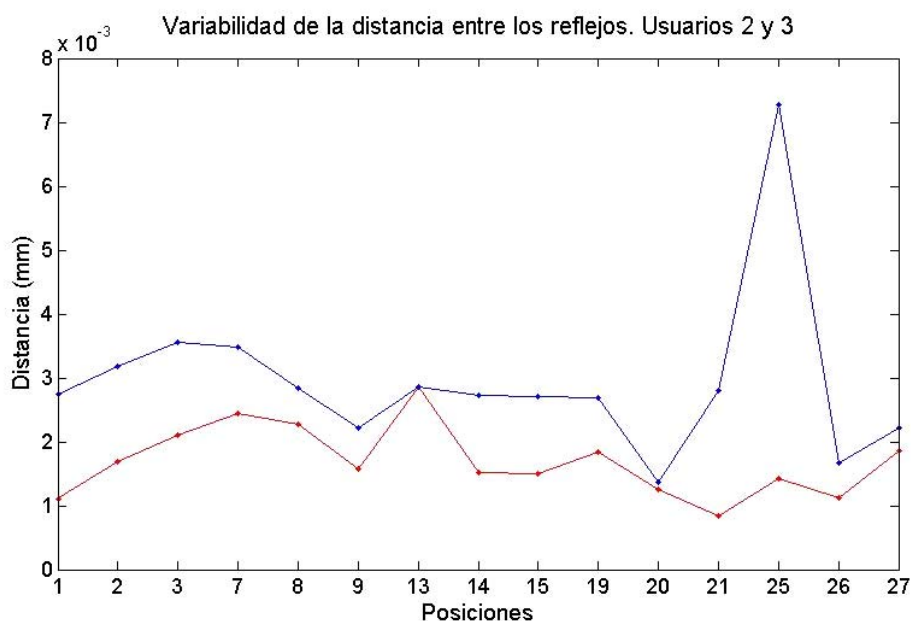


Fig. 7.46 Desviación estándar de la distancia entre los reflejos

En la primera figura (fig.7.45) se muestra el comportamiento de la distancia entre los reflejos de los usuarios 2 y 3 al avanzar de posición. Se pueden diferenciar fácilmente en la gráfica las tres caras correspondientes a cinco posiciones consideradas en la misma distancia en el eje Z. Sin embargo, además de ello se observa como las posiciones superiores del cubo están más lejos de la cámara ya que la distancia entre los reflejos es algo inferior. La variabilidad de los datos de esta distancia además es menor a las diferencias entre la distancia de los reflejos de cada una de las posiciones (ver fig.7.46). Es curioso que para el usuario 3 esta variabilidad aumente. (Los datos correspondientes a la posición 15 de este usuario pueden ser debidos a algún error de procedimiento durante la calibración).

### 7.4.2 Datos provenientes de un modelo matemático

Para complementar este estudio se decide realizar el mismo análisis que ha sido realizado en el apartado anterior esta vez utilizando datos de un modelo matemático creado por [7]. Únicamente trabajaremos con los parámetros fisiológicos correspondientes a un usuario medio.

- Distancia centro córnea – centro globo ocular: 5.55mm
- Distancia centro globo ocular – centro pupila: 10mm
- Radio de la córnea: 7.7mm
- Radio pupila: 3mm
- Coeficiente de refracción: 1.336
- Desviación angular eje óptico – eje visual:  $5^\circ$
- Distancia focal de la cámara: 100mm

A continuación se muestra la configuración y disposición de los elementos del sistema que se utilizaron para obtener los datos del modelo matemático que disponemos.

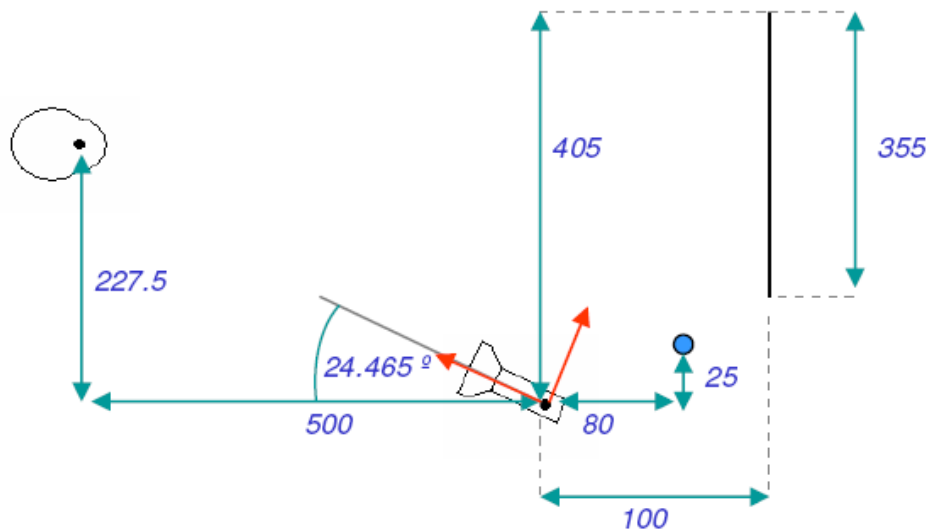
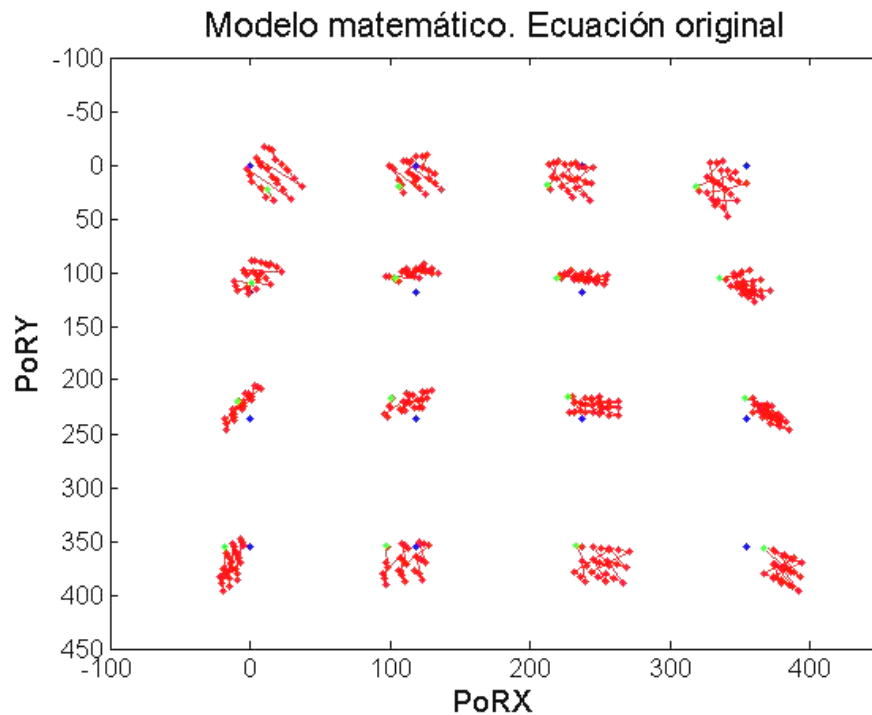


Fig. 7.47 Esquema de la disposición de los elementos del sistema

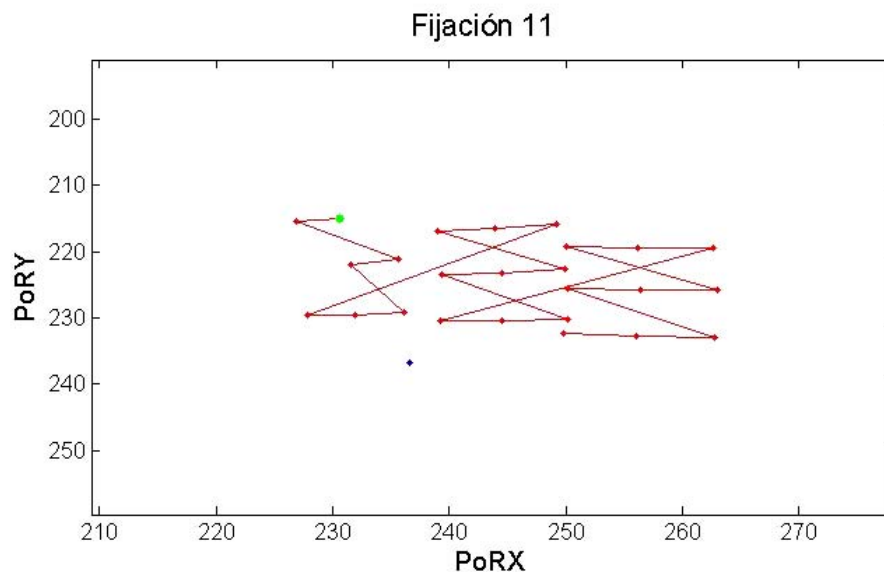
Contamos con la información de 25 de las 27 posiciones del cubo presentado en el subapartado 7.4, nos falta información de las posiciones 1 y 6.

Los datos del modelo matemático vienen dados en milímetros lo cual es indiferente para realizar todos los cálculos. Es decir, los coeficientes serán los mismos a los que obtendríamos al pasar los datos en distancia real a píxeles. Pero, al término de corrección de la ecuación fija habrá que aplicarle un factor de escala, solamente al término constante " $k$ " o la pendiente de la recta de corrección (ver ecuación 7.1). El factor será la división del término para pasar los píxeles a mm de la pantalla entre el término para pasar de píxeles a mm en la imagen de la cámara. (Ver apartado 4 para calcular el factor de escalado).

Primero realizaremos una estimación del  $PoR$  a partir de los datos del modelo matemático utilizando la ecuación de calibración original.



**Fig. 7.48** Estimación del  $PoR$ . Ecuación original



**Fig. 7.49** Estimación del  $PoR$ . Zoom en la fijación 11.

En el gráfico anterior (ver fig. 7.48) podemos observar el comportamiento de la estimación del  $PoR$  a medida que el usuario se desplaza en los tres ejes. La figura 49 muestra más claramente el comportamiento de la estimación media de una fijación ante desplazamientos del usuario en el eje  $Z$  el cual se repite en todas las fijaciones. Respecto a los datos obtenidos con usuarios reales, en este caso los desplazamientos en el eje  $Z$  no provocan tanta variación en la estimación del  $PoRY$ . En el caso del usuario medio de las pruebas realizadas en el estudio experimental al avanzar 4cm (al avanzar dos posiciones en dichas pruebas) únicamente en el

eje  $Z$  el error en  $Y$  medio aumenta 35 píxeles (ver figura 7.28) y en este caso al avanzar 5cm (la distancia entre posiciones) el error en  $Y$  no varía ni 2 píxeles. Además, estos resultados presentan mayor desviación ante desplazamientos en el eje  $X$  e  $Y$ .

Obviamente la aplicación de las nuevas ecuaciones de calibración propuestas no mejoraría la estimación del  $PoRY$  ante desplazamientos en el eje  $Z$  ya que no existe fuerte dependencia entre este desplazamiento y el error en  $Y$ .

Concretamente si aplicásemos la ecuación de corrección individual tendríamos que calcular la pendiente de la recta de corrección ( $k$ ) el cual estará cerca de cero. Recordemos que  $k$  se calcularía como la diferencia entre el error en  $Y$  de las posiciones 15 y 13 entre la distancia entre los reflejos de la posición 15 menos la de la 13. Por tanto este término será bajo ya que la diferencia entre el error en  $Y$  de las posiciones 13 y 15 es muy baja respecto al error en  $Y$  provocado por desplazamientos en el eje  $Z$  en casos reales. Al aplicar la ecuación con corrección fija aumentaríamos el error respecto a la ecuación original considerablemente ya que el error provocado por estos desplazamientos dista del caso real.

Esta diferencia entre la estimación del PoR de usuarios reales y de los datos del modelo matemático pueden ser debidos a que las condiciones en las que se han realizado las pruebas del estudio experimental como la distancia del usuario al sistema, las dimensiones de la pantalla utilizada, etc. es diferente de la distribución modelada matemáticamente.

Por último vamos a analizar el comportamiento de la distancia entre los reflejos ante desplazamientos en cada uno de los ejes y ver si se corresponde con la realidad, y a partir de ahí estudiar su comportamiento ante desplazamientos en el resto de los ejes. El objetivo último de esta prueba es tratar de intuir si el término corrector de la nueva ecuación de calibración afectaría o empeoraría el resultado ante desplazamientos del usuario en otros ejes que no fueran el  $Z$ . Para ello se van a mostrar la media y la desviación de la distancia entre los reflejos de tres posiciones para los desplazamientos en cada eje.

Desplazamiento en el eje Z		
Posiciones	Distancia entre los reflejos (mm)	
	Media	$\sigma$
13	0.32879	0.00055
14	0.27844	0.00038
15	0.23863	0.00027

Tabla VIII. Desplazamiento en el eje  $Z$

Desplazamiento en el eje X		
Posiciones	Distancia entre los reflejos (mm)	
	Media	$\sigma$
5	0.27781	0.00039
14	0.27844	0.00038
23	0.27787	0.00037

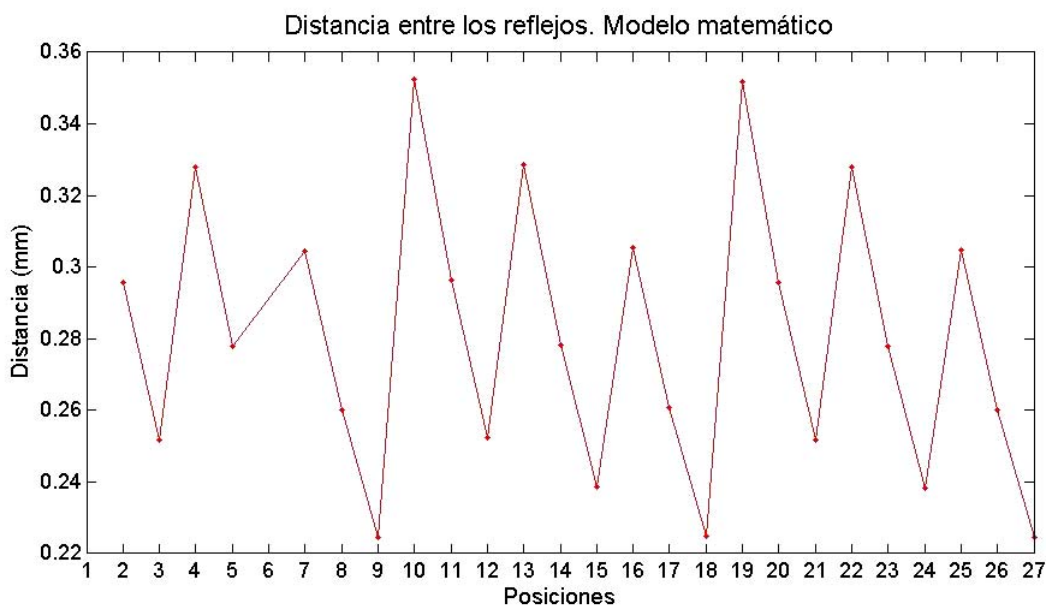
Tabla IX. Desplazamiento en el eje  $X$

Desplazamiento en el eje Y		
Posiciones	Distancia entre los reflejos (mm)	
	Media	$\sigma$
11	0.29627	0.000044
14	0.27844	0.00038
17	0.26068	0.00032

**Tabla X.** Desplazamiento en el eje Y

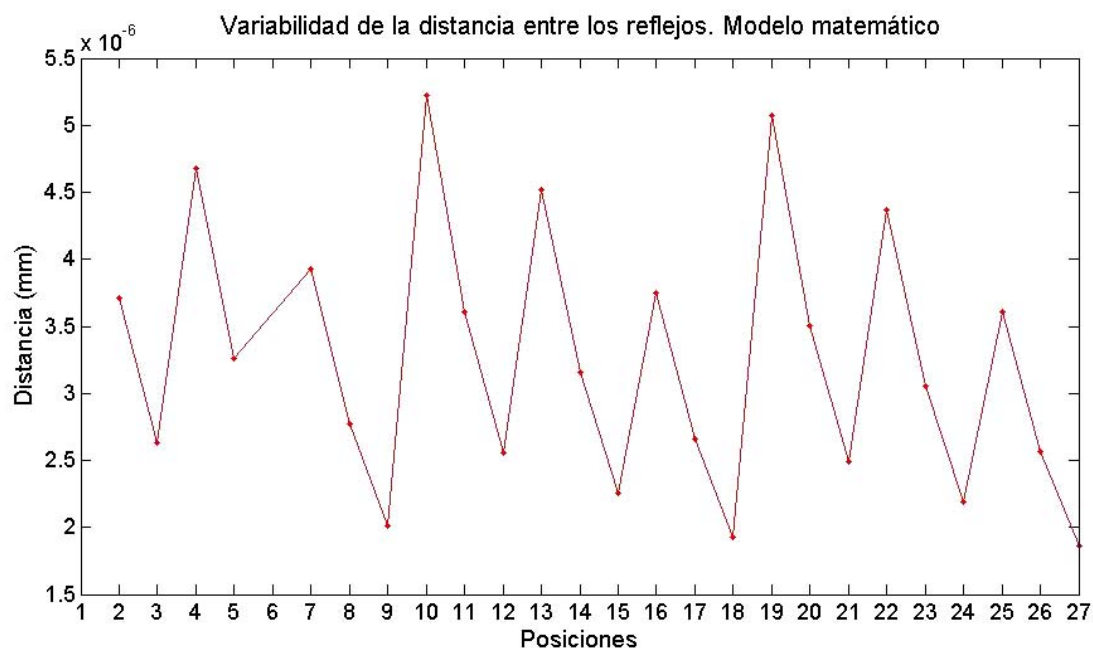
La información recogida en las tablas VIII, IX y X es un claro indicativo de que la distancia entre los reflejos principalmente varía ante desplazamientos en el eje Z. Además la información del modelo matemático nos permite comprobar la variación de esta ante desplazamientos entre los ejes X e Y. En el caso del eje X esta variación es muy baja en comparación con la variación existente en el desplazamiento en el eje Z, sin embargo, en el desplazamiento en el eje Y hay más variación respecto a las pruebas experimentales aunque no es tan significativa como en el caso del eje Z. En todas las posiciones observamos la desviación de la distancia entre los reflejos al mirar a toda la pantalla, es decir, cuando el usuario barre la pantalla con el ojo, la cual es muy baja respecto a las diferencias de esta variable en cada posición.

Por último mostraremos la distancia entre los reflejos media de cada posición junto con la desviación estándar en cada una de ellas (ver fig. 7.50 y 7.51). Esta representación nos permitirá comprobar si los resultados ofrecidos por el modelo matemático se alejan de la situación real o no.



**Fig. 7.50** Distancia entre los reflejos media de cada posición





**Fig. 7.51** Desviación estándar de la distancia entre los reflejos de cada posición

Aunque la falta de la información correspondiente a las posiciones 1 y 6 del cubo pueda confundir, en general la distancia entre los reflejos conforme avanzamos de un mismo punto desde la posición delantera hacia la más alejada la distancia entre los reflejos disminuye de forma casi lineal (ver fig. 7.50). Además claramente se observa que no todas las posiciones de una misma cara en el cubo están a la misma distancia de la cámara ya que debido a la orientación de la misma las posiciones más bajas se encuentran a menor distancia que las altas, por ello existe variación de esta distancia ante desplazamientos en el eje  $Y$  como veíamos en la tabla  $X$ . Sin embargo, esta variación no es tan pronunciada como la existente ante desplazamientos en el eje  $Z$ . La desviación estándar de la distancia entre los reflejos tiene el mismo comportamiento que presenta esta variable pero de magnitud mucho menor ya que conforme nos alejamos de la cámara aumentamos el error del sistema (ver fig. 7.51). En una posición más alejada el ojo será de menor tamaño en la imagen capturada por la cámara por lo que el error corresponderá a una mayor distancia real.

A pesar de que el comportamiento de la distancia entre los reflejos del modelo matemático y de los usuarios reales sea el mismo, los valores de esta variable no se corresponden ya que la distancia relativa entre las posiciones del cubo y la cámara en el caso del modelo matemático es inferior. Además al tratarse de posiciones diferenciadas en 5cm en el caso del modelo matemático y de 3cm en el real la comparación tampoco sería posible.

### 7.4.3 Conclusiones

Una vez comprobada la mejora de la estimación del  $PoR$  ante movimientos en el eje  $Z$  por último para validar las nuevas ecuaciones propuestas hemos comprobado que el resultado de estas no es contraproducente al efectuar movimientos en los otros dos ejes:  $X$  e  $Y$  mediante pruebas con 3 usuarios en las que estos deberán calibrar en un cubo de 27 posiciones y comprobar que los nuevos movimientos no perjudican la estimación del  $PoR$ .

De la misma manera se han estudiado los resultados de un modelo matemático. No obstante la estimación del  $PoRY$  apenas varía 5píxeles ante movimientos de 10cm en el eje  $Z$  y

en las pruebas realizadas con usuarios, ante desplazamientos de 7cm el error llega a aumentar 50píxeles utilizando la ecuación original. Por lo tanto, la aplicación de las nuevas ecuaciones a los datos de este modelo no da lugar ya que únicamente perjudicaría la estimación.

En las pruebas con usuarios es claro que ante movimientos en el eje  $Z$  las nuevas ecuaciones optimizan la estimación del  $PoR$  y la variación obtenida ante desplazamientos en los ejes  $X$  e  $Y$  es notoria, sin embargo, a la vista de la variabilidad del error estos cambios no son significativos.

## 8 Conclusiones

En este proyecto se va a tratar de aumentar la robustez de un sistema de eye-tracking video oculográfico basado en ecuaciones polinomiales ante desplazamientos del usuario. Para ello recurriremos a otra herramienta (FaceAPI), la cual permite disponer de la localización espacial del ojo. Mediante la sincronización de ambos sistemas se corregirá la ecuación de calibración original, dotando al sistema de una mayor precisión frente a desplazamientos del usuario. Hemos realizado este estudio en dos pasos:

### 1) Caracterización de FaceAPI

- FaceAPI nos ofrece información espacial de la posición y rotación de la cabeza asumiendo un error de 1cm al estimar la posición y un error de 3 grados al estimar la rotación. También ofrece información de la posición del ojo y de otras features de la cara.

- Hemos encontrado cierta imprecisión al calcular la posición del usuario en el eje  $Z$  a partir de la información de varios usuarios situados en el mismo lugar FaceAPI no considera que todos se encuentren en la misma posición cometiendo una diferencia entre cada usuario muy superior a 1cm. Más perjudicial aún es el hecho de que las medidas relativas a desplazamientos en este mismo eje de diferentes usuarios no son correctas.

- FaceAPI puede que trabaje con un modelo matemático interno ya que para obtener la localización espacial de un objeto a partir de un sistema sin calibrar de una única cámara, es necesario recurrir a un modelo matemático-estadístico previo para calcular la posición relativa entre el usuario y la cámara. Es posible que a partir de una serie de parámetros de la imagen (medidas antropométricas como la distancia entre los ojos, la nariz, etc.) se realice algún tipo de calibración interna del modelo. De este modo, la distancia entre el usuario y la cámara (eje  $Z$ ) corresponderá a aproximaciones realizadas a partir de dicho modelo, lo cual puede causar imprecisión en los resultados.

- Al estudiar el frame rate al que funcionaba por defecto FaceAPI descubrimos que no era suficiente para trabajar con Shakti, por ello se realizó un cambio de formato para utilizar una resolución de las imágenes de vídeo de 320 x 240 píxeles consiguiendo así el funcionamiento del sistema tiempo real (a 30 frames/segundo), manteniendo la precisión.

### 2) Estudio de la cuantía del error presentado por el sistema de eye-tracking y posterior corrección de la ecuación de calibración

- Se ha realizado un estudio de la cuantía del error del sistema de eye-tracking ante desplazamientos de los usuarios controlados y al mismo tiempo cuantificados por FaceAPI en el eje  $Z$  considerado como el movimiento más crítico.

- En este estudio han participado un total de doce usuarios para medir el error cometido por la ecuación de partida. En comparación con estudios anteriores [3], [4], [6], [9], el número de usuarios es muy elevado, dotando al estudio de gran relevancia práctica.

- Los resultados de estas pruebas muestran que el movimiento del usuario en el eje  $Z$  afecta fundamentalmente a la estimación del  $PoRY$ . Además, el error en  $Y$  varía linealmente conforme nos alejamos de la posición original en la que se calibró el sistema (ver fig. 7.6). El error de la estimación del  $PoRY$  en la posición central es de 18píxeles y en posiciones extremas de 68píxeles.

- A partir de estos resultados, se ha tratado de relacionar el error en el eje  $Y$  como consecuencia de los desplazamientos del usuario, con la información espacial proporcionada por FaceAPI. No obstante, el comportamiento de la distancia del usuario a la cámara no ha resultado satisfactoria, presentando un comportamiento poco adecuado para la caracterización de los desplazamientos en el eje de interés. En vista de estos resultados, se ha tratado de buscar otras soluciones que no dependan de esta variable para corregir la ecuación de calibración.

- El objetivo es hallar una variable que mantenga una relación aproximadamente lineal con la posición en  $Z$  y no varíe ante desplazamientos en otros ejes ni al cambiar la dirección de la mirada: la distancia entre los reflejos. Esta variable será el punto de partida para compensar el error producido por la ecuación de calibración con la distancia a la pantalla añadiendo a la ecuación de calibración  $PoRY$  un término proporcional a esta.

- La nueva ecuación corresponde con una recta de compensación (ya que el error en  $Y$  es proporcional a la distancia en el eje  $Z$ ), es decir, una constante que multiplica con la distancia de los reflejos más un término independiente que puede ser calculada a partir de la información de las pruebas o de forma independiente para cada usuario.

- Ambas correcciones implican un resultado parecido. La primera solución es poco menos precisa (comete un error en  $Y$  de 36 píxeles en posiciones extremas de las pruebas realizadas frente a los 32 que comete la segunda) no obstante la utilización de la segunda ecuación implica aumentar el tiempo de calibración. Este aumento del tiempo de calibración es necesario para calcular el término de corrección de la ecuación ya que además de la calibración en la posición central el usuario deberá moverse hacia delante y hacia atrás en el eje  $Z$  y volver a calibrar, esta vez solamente en 4 puntos (o fijaciones).

- Para cerrar el proyecto se ha comprobado que el término añadido a las nuevas ecuaciones no perjudica la estimación del  $PoR$  ante desplazamientos del usuario en los ejes  $X$  e  $Y$ . Esto ha sido comprobado de forma empírica con 3 usuarios mediante el estudio en 27 localizaciones espaciales formando un cubo. De forma complementaria se ha recurrido a un modelo matemático de un sistema video-oculográfico, el cual nos ha permitido verificar los resultados desde un punto de vista teórico.

## 9 Líneas futuras

- El primer paso sería la integración de la ecuación de compensación en un sistema real como es Shakti.

- Una vez corregida la ecuación de calibración permitiendo al usuario el libre movimiento en el eje  $Z$  parece claro que el siguiente paso sería intentar prever la tendencia del error ante movimientos de los usuarios en los otros dos ejes,  $X$  e  $Y$  e intentar modificar la ecuación de calibración para permitir el libre movimiento del usuario en los tres ejes.

Para ello podríamos recurrir nuevamente a FaceAPI para calcular la posición en los eje  $X$  e  $Y$  de la cabeza ya que parece que el comportamiento de este sistema ante estos desplazamientos es más preciso. Podría realizarse un estudio equivalente al que se ha realizado en este proyecto para analizar el comportamiento del eje  $Z$  y estudiar la utilidad potencial de la posición en  $X$  e  $Y$  si se tratara de corregir el error en esos ejes de el sistema de eye-tracking.

- Además también podría integrarse un sistema estéreo para una localización del ojo del usuario en 3D más precisa que la ofrecida por FaceAPI.

- En el caso de que se utilizase FaceAPI incluso para determinar la posición del ojo en los ejes  $X$  e  $Y$ , un estudio más detallado pasaría por tratar de establecer una relación entre la distancia en el eje  $Z$  y el resto de los parámetros antropométricos para poder intentar corregir la ecuación a partir de la información de la posición de la cabeza en los tres ejes.

-Es muy posible que además de los parámetros considerados en este trabajo como potenciales candidatos a formar parte del modelo, como la posición del usuario en el espacio, existan otras muchas posibilidades. Diferentes criterios para la normalización de parámetros como se intentó hacer con la normalización del vector  $V_y$ .

- Realizar un estudio experimental más completo del comportamiento del sistema ante desplazamientos del usuario en un cubo, ya que dada la complejidad del procedimiento únicamente se han estudiado 3 usuarios.

- Tratar de demostrar empíricamente cómo ciertos movimientos o desplazamientos del usuario podrían descomponerse como combinación de otros, ya que las pruebas que se realizaron en este proyecto no fueron concluyentes.

## **Bibliografía**

- [1] Juan J. Cerrolaza, A. Villanueva, R. Cabeza, Taxonomic study of Polynomial Regressions Applied to the Calibration of a Video-Oculographic System.
- [2] D. Li, D. Winfield, D. J. Parkhurst, Starburst: A hybrid algorithm for video-based eye-tracking combining feature-based and model-based approaches, in: Computer Vision and Pattern recognition, 2005 IEEE Computer Society Conference on, Vol. 3, 2005.
- [3] T. Ohno, N. Mukawa, A free head, simple calibration, gaze tracking system that enables gaze-based interaction, in: Proceedings of the Eye Tracking Research & Applications Symposium, San Antonio (TX), 2004, pp. 115–122.
- [4] D. Beymer, M. Flickner, Eye gaze tracking using an active stereo head, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, Wisconsin, 2003, pp. 451–458.
- [5] Andrew T Duchowsky, Eye Tracking methodology Theory and Practice, Springer
- [6] E. Guestrin, M. Eizenman, General theory of remote gaze estimation using pupil center and corneal reflections, IEEE Transactions on Biomedical Engineering 53 (6) (2006) 1124–1133.
- [7] A. Villanueva, Mathematical models for video-oculography, Ph.D. thesis, Public University of Navarra (2005).
- [8] J.J Cerrolaza, Proyecto fin de carrera. Universidad Pública de Navarra
- [9] S. W. Shih, J. Liu, A novel approach to 3-D gaze tracking using stereo cameras, IEEE Transactions Systems Man and Cybernetics Part-B 34 (1) (2004) 234–245.
- [10] J. Chen, Y. Tong, W. Gray, Q. Ji, A robust 3D eye gaze tracking system using noise reduction, in: Proceedings of the Eye Tracking Research and Applications Symposium, Savannah, Georgia, USA, 2008.
- [11] [http://www.eyethink.org/publications\\_assets/EyeTrackingEBBE.pdf](http://www.eyethink.org/publications_assets/EyeTrackingEBBE.pdf)
- [12] [http://www.ergoestudio.com/articulos/articulos/diferentes\\_aplicaciones\\_et.php](http://www.ergoestudio.com/articulos/articulos/diferentes_aplicaciones_et.php)
- [13] T. Ohno, N. Mukawa, A. Yoshikawa. FreeGaze: A Gaze Tracking System for Everyday Gaze Interaction. In *ACM Eye Tracking Research and Applications*, New Orleans, LA, March, 2002, pp. 125-132.
- [14] C. Morimoto, A. Amir, and M. Flickner, “Detecting eye position and gaze from a single camera and 2 light sources,” in *Proceedings. 16th International Conference on Pattern Recognition*, 2002, pp. 314–317.
- [15] Zhiwei Zhu and Qiang Ji, Eye Gaze Tracking Under Natural Head Movements. Department of ECSE, Rensselaer Polytechnic Institute, Troy, NY, 12180
- [16] Y. Ebisawa, M. Ohtani, and A. Sugioka, “Proposal of a zoom and focus control method using an ultrasonic distance-meter for video-based eye-gaze detection under free-hand

condition,” in *Proceedings of the 18th Annual International conference of the IEEE Eng. in Medicine and Biology Society*, 1996.

[17] C.H. Morimoto, D. Koons, A. Amir, and M. Flickner, “Frame-rate pupil detector and gaze tracker,” in *IEEE ICCV’99 Frame-rate Workshop*, 1999.

[18] LC Technologies Inc, “The eyegaze development sytem, <http://www.eyegaze.com>,” .

[19] C.H. Morimoto and M. R.M. Mimica, “Eye gaze tracking techniques for interactive applications,” *CVIU, special issue on eye detection and tracking*, 2005.

[20] T.E. Hutchinson, K.P. White Jr., K.C. Reichert, and L.A. Frey, “Human-computer interaction using eye-gaze input,” in *IEEE Transactions on Systems, Man, and Cybernetics*, 1989, pp. 1527–1533.

[21] R.J.K. Jacob, “Eye-movement-based human-computer interaction techniques: Towards non-command interfaces,” 1993, pp. 151–190, Ablex Publishing corporation, Norwood, NJ.

[22] N. Ramanauskas, Calibration of video-oculographical eye-tracking system, ISSN 1392-1215 Electronics and Electrical Engineering (8) (2006) 65–68.

[23] K. P. White, T. E. Hutchinson, J. M. Carley, Spatially dynamic calibration of an eyetracking system, *IEEE Transactions on Systems, Man, and Cybernetics* 23 (4) (1993) 1162–1168.

[24] J. Merchant, R. Morrisette, J. Porterfield, Remote measurement of eye direction allowing subject motion over one cubic foot of space, *IEEE Transactions on Biomedical Engineering* 21 (4) (1974) 309–317.

[25] X. L. C. Brolly, J. B. Mulligan, Implicit calibration of a remote gaze tracker, *Conference on Computer Vision and Pattern Recognition Workshop* 8 (2004) 134.

[26] Jacob, R. J. K. (1991). The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look At is What You Get. *ACM Transactions on Information Systems*, 9(3), 152-169.

[27] Javal, E. (1879). Essai sur la Physiologie de la Lecture. *Annales D'Oculistique*, 81, 61-73.

[28] Huey, E. B. (1898). Preliminary Experiments in the Physiology and Psychology of Reading. *American Journal of Psychology*, 9(4), 575-886.

[29] Delabarre, E. B. (1898). A method of recording eye-movements. *American Journal of Psychology*, 9(4), 572-574.

[30] Dodge, R., & Cline, T. S. (1901). The angle velocity of eye movements. *Psychological Review*, 8(2), 145-157.

[31] Dodge, R. (1906). Recent studies in the correlation of eye movement and visual perception. *Psychological Bulletin*, 3(3), 85-92.

[32] Buswell, G. T. (1935). *How people look at pictures: a study of the psychology and perception in art*. Oxford, England: Univ. Chicago Press.

- [33] Fender, D. H. (1964). Contact lens stability. *Biomedical and Scientific Instrumentation*, 2, 43-52.
- [34] Yarbus, A. L. (1965). *Role of eye movements in the visual process*. Oxford, England: Nauka.
- [35] Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3), 372-422.
- [36] Taylor, S. E. (1971). The dynamic activity of reading: A model of the process. In *EDL Research and Information Bulletin* (pp. 9). New York: McGraw-Hill.